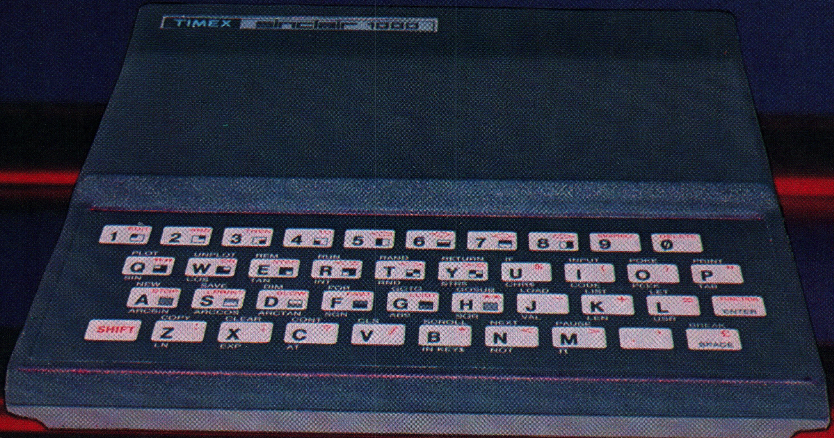
McGraw-Hill/VTX Series

>>$8.95

CRUNCHERS

21 SIMPLE GAMES for the TIMEX/SINCLAIR 1000 2K



Yin Chiu / Henry Mullish

CRUNCHERS

21 Simple Games for the Timex/Sinclair 1000 2K

Yin Chiu and Henry Mullish

McGraw-Hill Book Company

New York St. Louis San Francisco Auckland Bogotá
Guatemala Hamburg Johannesburg Lisbon London Madrid
Mexico Montreal New Delhi Panama Paris San Juan
São Paulo Singapore Sydney Tokyo Toronto

As used in this book, the terms "Timex/Sinclair 1000" and "T/S 1000" refer to the Timex/Sinclair 1000, a computer manufactured and sold by the Timex Corporation. Timex/Sinclair 1000 is a registered trademark of the Timex Corporation.

CRUNCHERS: 21 Simple Games for the Timex/Sinclair 1000 (2K)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 SEM SEM 8 7 6 5 4 3 2

ISBN 0-07-010831-5

Library of Congress Cataloging in Publication Data

Chiu, Yin, 1961-
"Crunchers" : 21 simple games for the Timex Sinclair 1000 (2K)

1. Games—Data processing. 2. Timex 1000 (Computer)
I. Mullish, Henry. II. Title.
GV1469.2.C47 1983 794 82-20334
ISBN 0-07-010831-5

Design and Production by Keithley and Associates, Inc.
Cover Photography by Paul Silverman
Illustrations by John Killgrew

Preface

When the Timex Sinclair 1000 was released in the summer of 1982, a barrier of sorts was broken. For the first time, a moderately powerful computer sold retail for under $100. No longer could the argument be made that computers were only for the wealthy. With the release of the T/S 1000, computers came within reach of the vast majority of the population, including students. Almost overnight, the computing power of large institutions of the 1950s was brought into the homes of "the people." Indeed, the T/S 1000 may well be termed (as was the Volkswagen) "the people's machine."

One of the aims of this book is to meet the unprecedented demand for video arcade games, which have gripped the imagination of millions of people. The book contains twenty-one original games programmed in the T/S 1000's version of the BASIC language. Not only will you be able to amuse yourself and your friends with these games once they are entered into your machine, but you will be able to follow the logic on which they are based, for explanations accompany each of the programs. And more than that—we also suggest ways in which you can easily amend the programs in order to personalize them to better fit your own desires.

Great care has been taken to print the programs in as

clear a fashion as possible, and none of the programs requires more than the 2K of random access memory (RAM) that is built into the computer. Despite the restrictions imposed by the size of available memory, you will be surprised at the wide range of games that are possible.

For those users who have purchased the additional 16K memory, suggestions are made at the end of many of the chapters on how to improve the programs and exploit the extra memory. And for the benefit of the novice, very explicit keyboarding instructions are given on how to enter programs, how to edit, how to run programs, how to save programs on a tape recorder, and how to load programs into the computer from tape.

The task of writing games for a computer is not an easy one. It is our wish that after you have had an opportunity to become familiar with the programs you will also examine them carefully and pick up some hints that might help you to write your own. This is an intellectual challenge that should not be minimized. Whatever expertise is gained in this endeavor can be put to direct advantage in regular programming, whether the discipline be mathematics, physics, chemistry, finance, or engineering.

If this is your first attempt at game playing on a computer, welcome, and lots of luck!

Yin Chiu
Henry Mullish

Contents

Chapter 0 GETTING STARTED

- How to Use This Book
- Setting Up the Computer
- Keyboarding Hints

Even if you are already experienced in using the Timex/Sinclair 1000 computer, you should take a few minutes to read this chapter. The T/S 1000 is remarkably powerful for so inexpensive a machine, but to use it to its best advantage you must be familiar with its unique dialect of BASIC, perhaps the most widely used computer language in the world, and you must be familiar with the peculiarities of the T/S 1000's keyboard, which is organized in such a way as to make keyboarding as painless and efficient as possible.

A brief reading of the points covered in this chapter, along with the instructions that appear in the *Timex User Manual* that came along with your computer, should enable you to set up the computer and enter the first program within one hour.

How to Use This Book

At the heart of this book are the twenty-one games programs, with the first program listed in Chapter 1 and the

twenty-first listed in Chapter 21. We strongly recommend that you enter the first four programs before jumping around in the sequence to select others. We make this recommendation because the first four programs are very clearly annotated, and contain explanations of many of the typographic conventions that we use in this book. In program listings following Chapter 4, we assume that you have mastered the basics of the keyboard, so the annotations are much fewer and further between.

Each chapter in this book begins with an explanation of the game that you will get when you run the program. You can get an idea of just what the game consists of by reading this section, called simply "The Game." Following the description of the game is the actual listing of the program, and in Chapters 1, 3, and 4 the listing is followed by "Keyboarding Notes," which are meant to provide the absolute beginner with instructions on how to manipulate the T/S 1000 keyboard.

Chapter 1 also contains instructions on how to edit a program and run a program. Neither of these activities is hard to do, but if you have done neither before, you should read these sections carefully.

All the chapters contain two additional sections: "Points of Interest" and "Possible Adaptations." The material under "Points of Interest" is intended as an explanation of the logic behind certain lines and groups of lines in the program. If you already have a pretty good grasp of BASIC, you can refer to these "points" if you want to analyze the logic of the programs. If you are an absolute beginner, these "points" will be somewhat less helpful but by studying them in conjunction with the program you will begin to get an inkling of how BASIC works.

The "Possible Adaptations" at the end of each chapter either give specific instructions on how to alter a program (for the absolute beginner) or *suggest* ways in which a

program can be customized (for more advanced readers). Note that some (but not all) of the adaptations require more than the 2K RAM built into the T/S 1000. If you haven't purchased the 16K RAM expansion pack, you will not be able to make all the modifications.

Understanding a program is a serious intellectual challenge, but it can also be a lot of fun. We hope that you will use the instructions in this book to tinker with and revise and expand and customize these programs. Before long you'll realize that you are truly "getting into" these programs, and in a very real sense, you'll be making them your own.

Setting Up the Computer

Instructions for setting up the T/S 1000 appear in Chapter 1 of the *Timex User Manual* (page 5). These instructions are fairly straightforward, and you should not have very much trouble following them. We have found, however, that a few additional adjustments to the setup can make your system even easier to use.

1. *The cable connecting your computer to the TV:* The manufacturer has supplied a 4-foot cable that connects your computer and the TV. We've found that in terms of convenience and comfort this cable is much too short, so we recommend that you replace the 4-foot cable with a 6- or 8-foot cable. (Most electronics or television and radio stores will have such cables in stock, and one shouldn't cost you more than a few dollars.)

For the most part the T/S 1000 produces a TV image with black characters on a white background, so the screen is quite bright. With the 4-foot cable supplied by the manufacturer, we found ourselves sitting almost on top of the screen, and after a few hours of work we were suffering from eye strain and beginning to feel a little bit spaced out. With a

longer cable, we were able to move the TV to the far end of a table. In such a setup, we still found the image on the screen to be perfectly readable, and it was certainly more restful for our eyes.

2. *The channel switch on the bottom of the computer:* At times when we've been programming or playing a game on the T/S 1000, we've slid the computer a few inches across the table to let another person take the controls. Sometimes when we've done this, we've completely lost the image on the screen. A little investigative work turned up the answer: the channel-selection switch on the bottom of the computer is supposed to be recessed into the machine, but on some of the computers we've dealt with it is not recessed enough. As a result, sliding the computer across a surface can throw the switch from one channel to the other, and there will be a temporary loss of the picture. This is not something to worry about. By simply flicking the switch back to the proper channel position, you should find that the image comes back onto the screen with no loss of whatever has been stored in memory. If this problem plagues your machine, you might want to consider "locking" the channel selection switch in place with a piece of tape.

We have also found that the manufacturer's advice about whether to use channel 2 or channel 3 does not always seem to be the best advice. Of the three machines we tested, two worked best on channel 3 and the third worked best on channel 2. Since all three machines were tested in New York City, where the local CBS-TV affiliate broadcasts on channel 2, we assumed that channel 3 would work best for us. But this was not the case for two of the three machines we tested. We can't explain this fact, but we want you to be aware of it.

3. *Setting up your tape recorder:* In saving programs and loading programs, we've found it best to have the

"volume" control of the tape recorder at or near its loudest setting, as mentioned in the *Timex User Manual.* Similarly, we've had the best results in loading and saving when the "tone" control is at its highest setting—toward "treble" rather than "bass." Furthermore, if your tape recorder has both "monaural" and "stereo" modes, use the "monaural" setting. If a stereo setting is all that is available, you should turn the "balance" control all the way to the left or all the way to the right and leave it there for both loading from and saving to the tape recorder.

The *Timex User Manual* also contains a very clever suggestion about how to store programs in such a way that they are easy to find: see page 14 of the manual for instructions on how to insert a voice announcement preceding each of the programs you store on tape. Be sure, however, to disconnect the microphone before you save the actual program from the computer to the tape.

Three further points: First, we've found that it pays to use a fairly high quality recording tape. Second, it is a good idea to make a backup copy of each program, preferably on a completely different cassette. And third, you should record programs on one side only of any tape, since if both sides are used, the signals on one side may interfere with those on the other.

4. Perhaps the greatest shortcoming of the T/S 1000 is its keyboard. Unless your hands are the size of Elliot's or E.T.'s, you'll have to take great care to hit the right key.

For maximum comfort, convenience, and accuracy of keyboarding, keep the keyboard well lighted and (as we have already mentioned) a good distance away from that glaring screen. With a few minutes of practice, you'll find that the keyboard is manageable.

Keyboarding Hints

If you haven't already had some experience with the

T/S 1000's keyboard, you should read this section carefully. As you read it, you may want to refer to the illustration of the keyboard in Appendix C at the end of this book.

1. When you are typing in the number 1, be sure to hit the 1 key and not the L key. The 1 key is located on the top row of the keyboard at the extreme left.

2. Don't confuse the number 0 with the letter O. In program listings in this book and on the screen of your TV, you'll notice that zero is printed with a slash through it, like this: Ø.

3. The letter I and the number 1 may also look alike to you as you read the program listings, but you must be sure to distinguish between them. The letter I is shown in the programs this way: I. The number 1 is listed in the programs this way: 1.

4. In some of the programs we make use of an underscore (_) to indicate that you should leave a space (by hitting the SPACE key). Do not attempt to key in this underscore, because you will not find it on the T/S 1000 keyboard. We make use of this symbol only within quotation marks in the program listing because it is only within quotation marks that inserting the proper spaces may be crucial.

5. In this book, the only quotation marks we use are the ones located on the P key. (For *none* of our twenty-one programs should you use the other quotation marks, which are located on the Q key.)

6. All words printed in **BOLDFACE** in the program listings are called keywords. Each of these words is accessed by a single keystroke. If you try to type in one of these words letter-by-letter, the computer will either not accept them or will not run the program correctly. So whenever you see a

boldface word in a listing, be sure to access it with a single keystroke. (A list of the T/S 1000's keywords and how to access them is contained in Appendix A at the back of this book.)

7. All of the following mathematical symbols are also accessed with a single keystroke:

- <> which means "not equal to" and is found on the T key.
- >= which means "greater than or equal to" and is found on the Y key.
- <= which means "less than or equal to" and is found on the R key.

8. Keep in mind that any given key on the T/S 1000 keyboard may have as many as five meanings, depending on the *mode* you are in. The mode is indicated by a cursor, as follows:

- K which indicates the keyword mode. When you are in this mode, you will access the word printed in white above the key.
- L which indicates the letter mode. When you are in this mode, you will access the "primary" (dark black) character on the key.
- F which indicates the function mode. When you are in this mode, you will access the word printed in white below the key.
- G which indicates the graphics mode. If you hit a key in this mode, the "primary" (dark black) character will appear in inverse video. (This means that the character will appear on the screen as a white number or letter printed in a small black box.) If you hold down the SHIFT key while in the graphics mode, some of the keys will give you the small graphics symbol shown at the lower right corner of the key. For a list of these graphics

symbols and how to access them, see Appendix B at the end of this book.

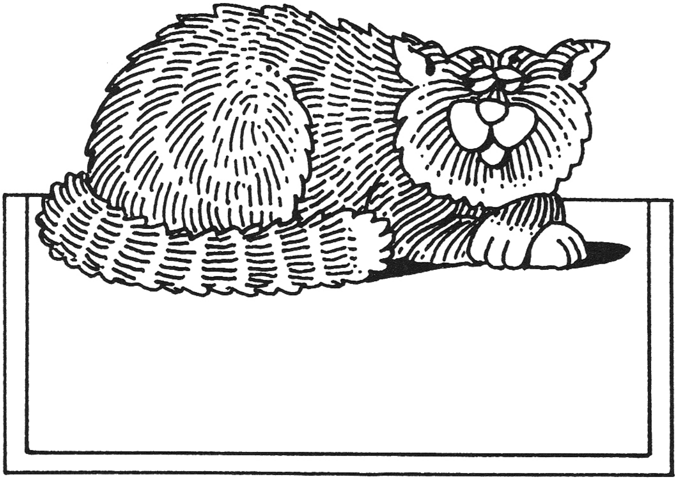
Chapter 1 COPY CAT

- The Game
- Program 1: COPY CAT
- Keyboarding Notes
- Editing the Program
- Running the Program
- Points of Interest
- Possible Adaptations

The first game we will present is called "Copy Cat." We've chosen to present it first not because it is the most interesting game in this book, but because it is one of the easier programs to understand. Whether you are a complete novice to programming or an old hand at it, the chances are good that you will derive a good deal of useful information from our step-by-step discussion of this program.

The Game

The object of "Copy Cat" is to type the letter that appears in the middle of the screen as soon as possible after it has been displayed. If you are successful in typing the letter correctly in the amount of time allotted, a point is added to your score automatically. Your score is then displayed at the



bottom of the screen, after which the next letter is presented. However, the time allowed for a response after each successive attempt is shortened. As soon as you either make a mistake in responding or take more time than is allowed, the game is terminated, indicating which letter was missed, and what, if any, the total score was.

What follows is a listing of the program. Take a few minutes now to look it over and then follow the explicit step-by-step instructions that appear just after the program listing. Within a few minutes you'll find that you are on your way to becoming an experienced T/S 1000 keyboarder.

Program 1: COPY CAT

```
1ØØ LET X$="ASDFGHJKL"
1Ø5 LET T=Ø
11Ø LET N=Ø
```

See the "Keyboarding Notes" that begin on page 11.

```
12Ø PRINT AT 3,12;"COPY CAT"
125 LET Y$=X$(INT (RND*9+1))
13Ø LET N=N+1
14Ø PRINT AT 11,15;Y$
15Ø PAUSE 15Ø -N
16Ø PRINT AT 11,15;"_"
17Ø IF INKEY$=Y$ THEN GOTO 2ØØ
18Ø PRINT AT 19,5;"YOU_MISSED;_IT_WAS_
    (";Y$;")"
19Ø STOP
2ØØ LET T=T+1
21Ø PRINT AT 21,15;T
22Ø GOTO 125
```

That underscore within quotation marks indicates a space.

Keyboarding Notes

There is a spare copy of the program "COPY CAT" at the back of this book. Tear it out now, and mark it with your own notes as you follow these steps:

1. Switch on the computer and be sure that the K cursor is displayed in the lower left corner of the TV screen.

2. As you are probably aware, all BASIC statements must begin with a statement or line number. Since we have arbitrarily assigned the line number 100 to the first line of the program, type in 100. You will note that the K cursor remains on the screen after 100 has been typed in.

3. After the statement or line number 100, you will want to type in the keyword **LET,** which is printed in **boldface** in this book's listing of the program. The boldface listing indicates,

as you learned in Chapter 0, that the word in question should be entered with a single keystroke, and not letter-by-letter.

4. Now type in the letter X, located near the left-hand corner of the bottom row of keys.

5. We now want to type in the symbol $. Notice that $ is printed in red, close to the top of the U key. This symbol is therefore accessed by using the SHIFT key, the one on the extreme left side of the bottom row of keys. With one finger hold down the SHIFT key and, at the same time, press the U key with a finger of the other hand. This operation must be done simultaneously. It will not work if you press these keys in succession—they must be depressed at one time. We will indicate this dual-key feature in the future by writing, for example, shifted-U.

6. The next symbol needed is the equal sign, =, which is found on the L key. You will notice that this is also printed in red, so once again the SHIFT key must be used. Remember to hold down the SHIFT key at the same time you depress the L key.

7. The double quote sign, ", is also printed in red, so it too will be accessed using the SHIFT key. (By the time you have typed in this rather short program you should become an expert on the SHIFT key!) Make sure to use the double quote sign—the one on the P key, not the paired quote sign on the Q key.

8. Now enter the sequence of letters ASDFGHJKL followed by a closing double quote sign (shifted-P). To send the whole line to the computer you must now press the ENTER key. If you have typed it in correctly, the line will be transferred to the top of the screen with the ▶ cursor appearing after the line number 100. (This symbol in reverse video is used on the computer as an edit cursor, about which we shall have something to say later on.) If you have not typed in the line

correctly, the S cursor, indicating a syntax error, may appear at the end of the line. In that case the T/S 1000's built-in checking system will not accept the line, and it will not move to the top of the screen when you hit the ENTER key. At this point the best thing to do may be to hit the DELETE key (shifted-0) as may times as necessary to remove the line. You can then start again. (This is actually performing an elementary editing function; a more complete discussion of editing appears later in this chapter.)

9. Lines 105 and 110 are straightforward—they are called assignment statements. They merely assign values to the variable specified on the left of the equal sign. Type in each and enter each into the listing by depressing the ENTER key. Again, each line will move to the top of the screen when it is entered. In line 120, the keyword **PRINT** is accessed by pressing the letter P while you are still in the K mode. The next thing to be typed is the keyword **AT**, which appears at the *bottom* of the C key. This means that it is a function, and can be entered only in the function mode. To get into the function mode, press the SHIFT and the ENTER keys simultaneously. Once this is done, the L cursor changes to the F cursor, signifying that you are now in the function mode. When the C key is pressed in the F mode, the word **AT** appears on the screen following the word **PRINT.** You will also notice that the L cursor has returned to the screen.

10. Now enter the number 3, followed by a comma, then the number 12, followed in turn by the semicolon and an opening quote sign. You will notice that the next two words in the listing, COPY and CAT, are in reverse video. In order to achieve this effect all that is necessary is to switch the computer into the graphics mode. To place the computer in the graphics mode simply press shifted-9. This displays the G cursor. Now type in the words COPY and CAT with one space between them.

11. To exit from graphics mode once you have entered COPY and CAT in reverse video, you may either hit the shifted-9 key again or the shifted-ENTER key. This permits the closing quote to be typed when the L cursor reappears.

12. Once you have sent line 120 to the top of the screen by depressing the ENTER key, you are ready to move on to line 125. Note that in order to key in **INT** and **RND,** you must be in the function mode. This means that you must access the F cursor twice by using the shifted-ENTER key for each of these words.

13. Lines 130, 140, and 150 are fairly straighforward. Key each one in and ENTER it now. In line 160, be sure to use the shifted-P key to access the double quote sign (") twice with one space between the quote signs, as indicated by the underscore. Do *not* try to key in this underscore, since it doesn't exist anywhere on the keyboard. Its only purpose is to indicate a space.

14. Line 170 is not difficult. Note, however, that the keyword **THEN** can be accessed in the L mode by simply hitting the shifted-3 key. Also notice that after **THEN** is keyed in, the K cursor reappears, making the keying of the next word, **GOTO,** a simple matter: simply hit the G key.

15. Once you've entered line 170, take a moment to stop and examine the way in which line 180 is listed in this book: note the four short lines or "underscores" that appear within the quotes at the end of the line:

```
18Ø PRINT AT 19,5;"YOU_MISSED;_IT_WAS_
    (";Y$;")"
```

These underscores cannot, as discussed in Chapter 0, be found on your T/S 1000 keyboard. Their only purpose here

is to indicate when and where the SPACE key should be hit when you are typing within quotation marks. Remember, within quotations you are typing what is called a "literal," and entering the correct number of spaces in the right places is often crucial in literals.

16. If you've gone ahead and typed in line 180, you may have already noticed that there are further differences between the listing in this book and what you see on the screen. First, let's look again at the listing as it appears in this book:

```
180 PRINT AT 19,5;"YOU_MISSED;_IT_WAS_
    (";Y$;")"
```

As you key this line onto the screen, however, it will appear as follows:

```
180 PRINT AT 19,5;"YOU MISSED; I
T WAS (";Y$")"
```

Obviously, the underscores that appear in the listing do not appear in the line as it appears on the screen. (See note 15.) In addition to that, however, you will note that the T/S 1000 automatically carries over the letter T of the word IT to a "new" line on the screen. (Actually, the T is still in line 180, so we call this "new" line a "turnover.") When you hit the ENTER key and send this line to the top of the screen, you will see still another change, as follows:

```
 180 PRINT AT 19,5;"YOU MISSED;
IT WAS (";Y$")"
```

Now the I of IT has been carried to the "turnover" to join its companion T. This occurs after you hit the ENTER key because the T/S 1000 has automatically shifted the statement number, 180, one space to the right, just in case you want to add any program lines numbered 1000 or more. This

is not of great importance in and of itself, but it *is* important for you to remember what we said in Chapter 0: The program listings in this book may show statement lines set up somewhat differently from what you see on the screen. This is not something to worry about, as long as you have taken great care to key in the proper space units within the literals. (And not made any mistakes in keying any other numbers, letters, and symbols, of course!)

17. Lines 190, 200, 210, and 220 are simple, and by now you shouldn't have much trouble with them. So key in each one and hit the ENTER key for each, and you'll be ready to . . . EDIT! We're sure that by now you're anxious to start running the game, but if you take just a few more minutes to double-check your work so far, you may save a lot of time in the future.

Editing the Program

Suppose that while typing in a line you make a mistake. Rest assured that this happens to *every* programmer, no matter how smart or how careful he or she is. So *when* it happens to you (notice we didn't say *if* it happens to you) don't panic—it's par for the course.

If you've completely forgotten to enter a line you can simply type that line at the bottom of the screen and hit the ENTER key. It will automatically insert itself in its proper place in the program. You can also retype a line that has been incorrectly entered; when you ENTER this retyped line, it will automatically "bump" the incorrect line from the program.

If when typing in a character you realize immediately that you hit the wrong key, the DELETE key (shifted-0) will erase the last character or keyword and reposition the cursor to the preceding location. This enables you to type in the correct character. The original one disappears for all time, and nobody need know.

Suppose, however, that a line has already been entered into the listing and you now decide that you want to change it in some way or other. If it is the current line that you want to change (the ⊵ cursor signifies which line is current), hit the EDIT key (shifted-1). This brings down a copy of the line to the bottom of the screen so that it may be changed. If the line in a listing does not have the cursor next to it, simply hit LIST and then the line number and ENTER. This will bring the line you want to the top of the screen with the ⊵ cursor next to it. Now it is the "current" line, and you can bring it down by hitting the EDIT key (shifted-1).

Once a line is at the bottom of the screen and ready to be edited, you can type shifted-8. This has the effect of moving the cursor to the right. (Shifted-5 moves the cursor to the left.) Stop at the appropriate place and make the changes you want. Then, when the ENTER key is pressed, the edited line returns to its proper place in the program.

Running the Program

Now most of the work has been done and it is time to execute the program. Make one last check of the program on your screen, comparing it carefully with the listing in this book. If they match, hit the R key, which in the K mode causes the **RUN** command to come up on the screen. Then, by hitting the ENTER key, you will start the program running.

Enjoy! And if you enjoy "COPY CAT" enough to want to save it, turn to Chapter 22 in this book, "Saving Your Programs on a Tape Recorder."

Points of Interest

Line 100: The set of letters from which the computer randomly picks a single letter is composed of A,S,D,F,G,H, J,K, and L. You will notice that these are the letters located on the third row of your computer keyboard. If you now

look at the listing of Program 1, you will notice that these nine letters are assigned to the string variable X$.

Lines 105 and 110: These lines set to 0 the T counter (for Total points) and the N counter (for the Number of frames to be used in the timing mechanism of the computer).

Line 120: This line uses two features that are very commonly found in games program. The first of these features is the instruction **PRINT AT.** This instruction allows whatever follows to be displayed on the screen at a specified point, as in this example from our program:

```
12Ø PRINT AT 3,12;"COPY CAT"
```

The 3 indicates that the first letter of the literal will appear at a point 3 lines from the top of the screen. The 12 indicates that the first letter of the literal will appear 12 spaces from the left-hand edge of the screen. The second feature used in line 120 and many other games programs is the inverse video function, which highlights white letters against a black background, thus setting them apart from other material displayed on the screen.

Line 125: This line selects one of the nine letters by using the random (**RND**) function that is built into the T/S 1000. The randomly selected letter is stored in the string variable named Y$.

Line 130: The frame count (where 1 frame is approximately equal to .02 second) is incremented by 1 in this line.

Line 150: This line contains a **PAUSE** which temporarily stops the execution of the program for the specified number of frames. (Again, there are about 50 frames per second.) This has the effect of "freezing" the display for the period specified, enabling the viewer to read what is there. At this point, the pressing of any key whatever discontinues the

effect of the **PAUSE** and permits the program to resume execution.

Line 160: A blank letter character is superimposed over the displayed letter, thereby giving the effect of the displayed letter being "flashed" on the screen for a limited period of time.

Line 170: If it is found in line 170 that the response matches the one shown on the screen, control is passed to line 200, where a point is added to the total score. This score is then displayed for the benefit of the player, and the next round is automatically initiated. However, if the keyed-in letter was incorrect or if it was not typed in time, the message YOU MISSED; IT WAS... is displayed, and the program is terminated.

Possible Adaptations

1. Notice that for each successful try, 1 is added to N in line 130. This affects the length of the pause in line 150 since it is specified to have a duration of 75 – N. If you think about this carefully, you will realize that the larger the value of N, the smaller this expression becomes and therefore the shorter the **PAUSE**.

2. The string in quotes in line 100 can of course be changed to any set of 9 characters you might choose.

Chapter 2 RANDOM

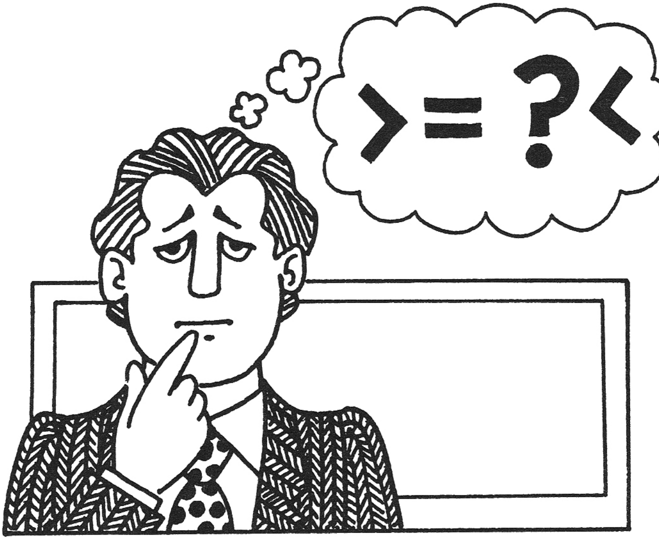
- The Game
- Program 2: RANDOM
- Points of Interest
- Possible Adaptations

The Game

This program causes the computer to generate a random number between 0 and 100. The user is asked to guess this number in the fewest tries possible. In this game the computer always displays a lower and an upper bound, between which the number is to be guessed. Initially, the display shows:

```
Ø<=X<=1ØØ
```

where 0 is the lower bound and 100 is the upper bound. For each guess made, a square dot is displayed near the bottom of the screen. As soon as a guess is typed in, a test is made to determine whether it matches the number randomly selected by the computer. Let us assume the computer randomly selects the number 38 and the player incorrectly guesses 76.



The computer adjusts the displayed bounds accordingly and will show the new bounds on the screen as shown:

Ø<=X<=76

which tells you that your guess was too high and you should now confine the range of your guess to a number between 0 and 76.

If your guess is correct, a message is displayed telling you the number of guesses that were made and the game terminates with a display of the following type:

38 == 38 == 38

This remains on the screen for a few seconds (during which time you may congratulate yourself) and the computer then selects another number for the next round. If you want to stop the game at any time simply type in **STOP** and then hit the ENTER key.

Program 2: RANDOM

```
100 LET G=Ø
11Ø LET X=INT (RND*1Ø1)
12Ø LET L=Ø
13Ø LET H=1ØØ
14Ø PRINT AT 3,8;"GUESS_MY_NUMBER"
15Ø PRINT AT 11,9;L;"_<=_X_<=_";H;"__"
16Ø INPUT N
17Ø LET G=G+1
18Ø FOR I=1 TO G
19Ø PLOT (G-1)*3+1,Ø
2ØØ NEXT I
21Ø IF N=X THEN GOTO 25Ø
22Ø IF N<X AND N>L THEN LET L=N
23Ø IF N>X AND N<H THEN LET H=N
24Ø GOTO 15Ø
25Ø PRINT AT 15,5;"YOU_GOT_IT_IN_";G;"_TRIES"
26Ø LET L=X
27Ø LET H=X
28Ø PRINT AT 11,9;L;"_==_";X;"_==_";H;"__"
29Ø PAUSE 15Ø
3ØØ CLS
31Ø GOTO 1ØØ
```

Watch it. That's the letter I that is equal to 1. See also line 200.

Points of Interest

Line 110: This line generates a random number between 0 and 100.

Lines 140 and 150: These lines set up the display to advise the user of what to do.

Line 160: This line accepts the user's guess.

Lines 170–200: These plot the square dots at the bottom of the screen, indicating the number of tries made.

Lines 210–230: These compare the guess to the computer's random number.

Lines 250–280: These lines display the result if the guess is correct.

Possible Adaptations

You could change the range of possible numbers from 0 to 100. A smaller range, from 0 to 10, for example, could be generated by replacing lines 110 and 130 as follows:

```
11Ø LET X=INT (RND*11)
13Ø LET H=1Ø
```

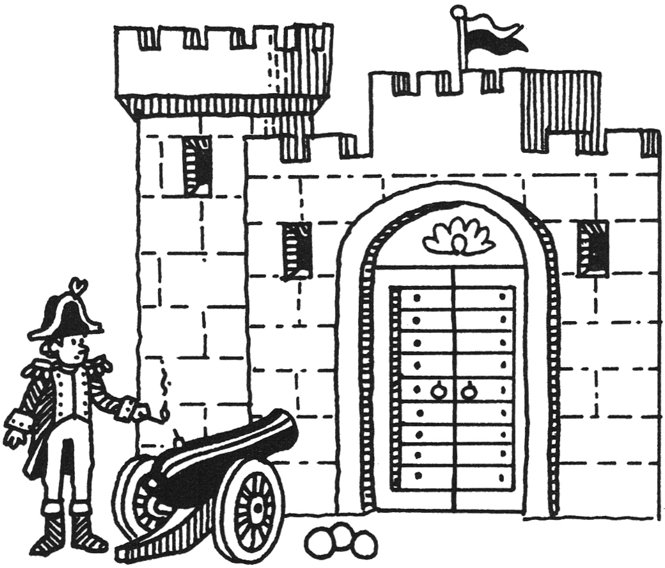
Chapter 3 GATES

- The Game
- Program 3: "GATES"
- Keyboarding Notes
- Points of Interest
- Possible Adaptations

The Game

"Gates" is a test not only of your physical dexterity but of your sense of timing. You are confronted with a pair of gates that continually open and close. The idea of the game is to shoot a bullet (in the form of the letter X) through the opening at the point at which the gates are closest together without actually being closed. If you shoot when the gates are either open too wide or are closed, you lose the point.

Any key on the keyboard may be pressed to shoot the bullet. If you are successful, you will see the bullet propelled from the bottom of the screen through the gate opening. Each successful attempt clocks up 100 points to the total score. For each unsuccessful attempt, you will lose 100 points. So long as you are successful, the game continues. More likely, you will fail at some point, and as soon as your score falls below –100, the game is terminated.



Program 3: "GATES"

```
100 LET I=12
105 LET P=0
110 LET J=15
120 LET K=J
130 FOR T=1 TO 6
135 IF T>3 THEN GOTO 170
140 LET J=J-1
150 LET K=K+1
160 GOTO 200
170 LET J=J+1
```

Again, that is the letter I that equals 12. See also lines 200, 210, 220, 230.

```
18Ø LET K=K-1
2ØØ PRINT AT I,J;"■"
21Ø PRINT AT I,K;"■"
215 IF INKEY$<>"" THEN GOSUB 3ØØ
22Ø PRINT AT I,J;"_"
23Ø PRINT AT I,K;"_"
24Ø NEXT T
25Ø GOTO 13Ø
3ØØ IF J<>14 THEN GOTO 4ØØ
3Ø5 FOR L=19 TO 2 STEP -1
31Ø PRINT AT L,15;"X"
315 PRINT AT L,15;"_"
317 NEXT L
32Ø LET P=P+1ØØ
33Ø PRINT AT 21,13;P;"___"
345 IF P<-1ØØ THEN STOP
35Ø RETURN
4ØØ LET P=P-1ØØ
41Ø GOTO 33Ø
```

Keyboarding Notes

1. Note the ■ symbol within literals at the end of line 200. It is accessed by first entering the G mode. (To get into the graphics mode, you hit shifted-9.) Then hit the SPACE key and the symbol will appear.

2. The <> symbol in line 215 is accessed with one keystroke, shifted-7. Do *not* use shifted-N and shifted-M.

Points of Interest

Lines 130–250: These control the movement of the gate.

Line 215: This line continuously scans the keyboard to determine if a shot has been fired.

Line 300: This line tests if the shot was successful.

Lines 305–330: The bullet is fired and 100 is added to the score for every hit, which is then displayed.

Line 345: This line tests whether the score is less than –100.

Line 400: If the shot misses, points are deducted by this line.

Possible Adaptations

1. In line 310, you may change the "X" to any other single character you might like to use to represent the bullet.

2. Rather than having a fixed value of 100 for each hit or a fixed value of –100 for each miss, you can have the computer generate a random value by making changes in the following lines:

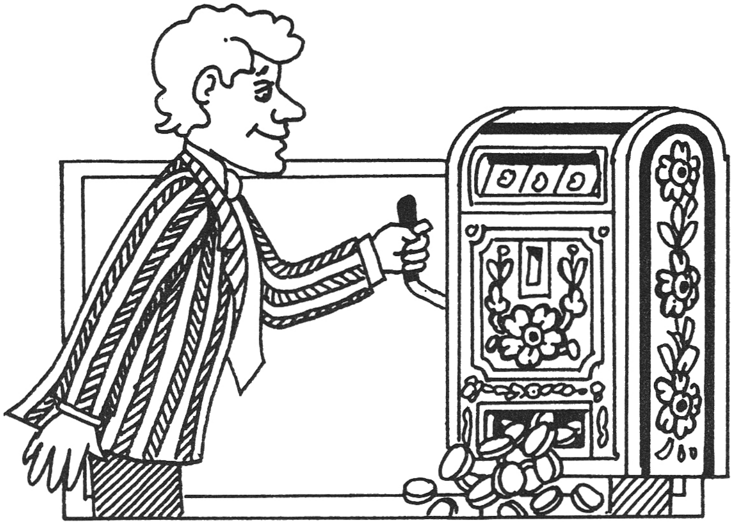
```
32Ø LET P=P+INT (RND*1ØØ)
4ØØ LET P=P-INT (RND*1ØØ)
```

Chapter 4 SLOT M

- The Game
- Program 4: SLOT M
- Keyboarding Notes
- Points of Interest
- Possible Adaptations

The Game

Most of us are familiar with slot machines, commonly referred to as "one-armed bandits." This program simulates the spinning windows that are characteristic of the slot machine. At the beginning of the game, the user is asked to type in the amount to be betted. This amount becomes the standard bet for the remainder of the game. If you watch the screen carefully, you will begin to see the first window spin. At any time you may stop the window from spinning by pressing any key. The moment the spinning of the first window stops, the second window begins to spin and the first one remains stationary. To stop the second window, press any key. After the second window has been stopped, the third window will start spinning. When this final window has been stopped (by pressing any key on the keyboard), the round is over and scores are awarded as follows:



- If all three windows display the same symbol, you have hit the jackpot, and the payout is five times the amount of the initial bet.
- If the first two match, the payout is two times the amount of the original bet.
- Otherwise you have lost the game.

The amount of winnings is then displayed at the bottom of the screen and a new round is initiated.

Program 4: SLOT M

```
5 LET B=Ø
6 PRINT "ENTER_BET"
7 INPUT R
8 CLS
```

```
10  LET I=8
20  LET J=4
25  PRINT AT 3,10;"SLOT MACHINE"
30  GOSUB 200
40  LET W1=T
50  LET J=13
60  GOSUB 200
70  LET W2=T
80  LET J=22
90  GOSUB 200
100 IF W1=W2 AND W2=T THEN GOTO 150
110 IF W1=W2 THEN GOTO 170
120 PRINT AT 19,9;"_YOU_LOSE"
125 LET Q=-1
130 GOTO 600
150 PRINT AT 19,9;"*_JACKPOT_*"
155 LET Q=5
160 GOTO 600
170 PRINT AT 19,9;"*_2_MATCH_*"
175 LET Q=2
180 GOTO 600
200 REM
210 PRINT AT I+1,J;"▛▀▀▀▜"
220 PRINT AT I+2,J;"▌█_█▐"
230 PRINT AT I+3,J;"▌_█_▐"
240 PRINT AT I+4,J;"▌█_█▐"
250 PRINT AT I+5,J;"▙▄▄▄▟"
```

See the "Keyboarding Notes" on page 33.

```
26Ø LET T=1
27Ø IF INKEY$<>"" THEN RETURN
3ØØ PRINT AT I+1,J;"█████"
31Ø PRINT AT I+2,J;"█▒█▒█"
32Ø PRINT AT I+3,J;"██_██"
33Ø PRINT AT I+4,J;"█▒█▒█"
34Ø PRINT AT I+5,J;"█████"
36Ø LET T=2
37Ø IF INKEY$<>"" THEN RETURN
4ØØ PRINT AT I+1,J;"█▀▀▀█"
41Ø PRINT AT I+2,J;"▌█_█▐"
42Ø PRINT AT I+3,J;"▌_█_▐"
43Ø PRINT AT I+4,J;"▌█_█▐"
44Ø PRINT AT I+5,J;"█▄▄▄█"
46Ø LET T=3
47Ø IF INKEY$<>"" THEN RETURN
5ØØ GOTO 21Ø
6ØØ LET B=B+Q*R
6Ø5 PAUSE 1ØØ
61Ø CLS
62Ø PRINT AT 21,8;"TOTAL_=_$_";B;
    "______"
63Ø PAUSE 1ØØ
64Ø GOTO 1Ø
```

Keyboarding Notes

1. Line 210 consists of five graphics symbols. To access these, first put the computer in the **G** mode by hitting the GRAPHICS key (shifted-9). Now that you are in the graphics mode, hit the following keys: shifted-E(▛), shifted-7(▀), shifted-7(▀), shifted-7(▀), and shifted-R(▜).

2. The literal in line 220 consists of five additional characters, four of which are accessed in the graphics mode. The central "character," however, is simply a space (symbolized by the underscore). It must be accessed from the **L** mode. To key it, leave the graphics mode (by hitting the shifted-9), hit the SPACE key once, and then reenter the graphics mode to key in the last two symbols.

Points of Interest

Lines 30–90: These set up the position of the windows.

Lines 100–180: These determine whether the game is lost or won and the amount of the payoff.

Lines 210–500: These display the spinning window and handle the polling of the keyboard.

Lines 600–620: The number of dollars won is calculated and the total is displayed.

Possible Adaptations

This program uses up almost all of the 2K memory available. Modifications are possible only if an additional RAM pack is added. Assuming this has been done, lines 620–640 may be amended to include a section in which the value of the bet for each round is entered before it starts.

Another possibility is to increase the number of symbols in each window to 4 or possibly 5. It should be stated, however, that this is a difficult change to implement; if you want to try it, you might be guided by examining lines 210–500 carefully.

Chapter 5 HIGH ROLLER

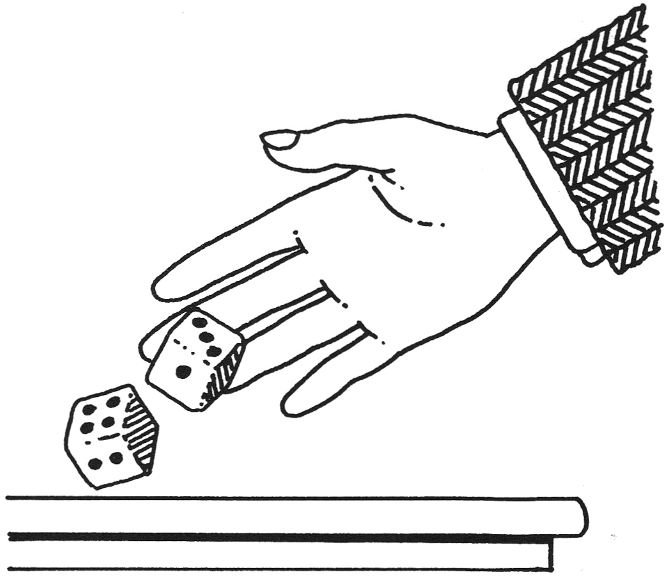
- The Game
- Program 5: HIGH ROLLER
- Points of Interest
- Possible Adaptations

The Game

This is a fast-paced graphics game similar in style to the familiar game of "craps." The rules for the game are:

- 7 or 11 on the first roll is a win.
- 2 or 3 on the first roll is a loss.
- Any other number on the first roll becomes your "mark." You must then roll the dice continually until the mark is rolled again. If a 7 is rolled before the mark, however, you lose.

Important Note: In order to contain the program within 2K of memory, it was necessary to avoid permitting the participant to interact. Once the program begins, it continues on its merry way by itself, permitting you only to observe the game in progress, and place whatever bets you please.



Program 5: HIGH ROLLER

```
1Ø LET I=9
15 LET M=Ø
2Ø LET J=11
22 PRINT AT 18,18;"__"
24 LET T=Ø
25 FOR K=1 TO 2
3Ø LET X=INT (RND*6+1)
4Ø IF K=1 THEN GOTO 6Ø
5Ø LET J=17
6Ø GOSUB (X+2)*1ØØ
7Ø LET T=T+X
```

```
 90 NEXT K
 95 LET M=M+1
100 PRINT AT 18,12;"VALUE_";T
103 PAUSE 100
105 IF M<>1 THEN GOTO 150
110 IF T=7 OR T=11 THEN GOTO 200
120 IF T=2 OR T=3 THEN GOTO 280
130 IF M<>1 THEN GOTO 150
140 LET H=T
143 PRINT AT 20,12;"MARK_";H
145 GOTO 20
150 IF H=T THEN GOTO 200
160 IF T=7 THEN GOTO 280
170 GOTO 20
200 PRINT AT 16,12;"YOU_WIN"
205 GOTO 285
280 PRINT AT 16,12;"YOU_LOSE"
285 PAUSE 100
286 LET M=0
287 PRINT AT 20,12;"_______"
288 PRINT AT 16,12;"________"
290 GOTO 20
300 PRINT AT I,J;"█████"
310 PRINT AT I+1,J;"█▛▀▜█"
320 PRINT AT I+2,J;"█▌_▐█
330 PRINT AT I+3,J;"█▙▄▟█"
340 PRINT AT I+4,J;"█████"
```

```
35Ø RETURN
4ØØ PRINT AT I,J;"▛▀███"
41Ø PRINT AT I+1,J;"▌_███"
42Ø PRINT AT I+2;J;"█████"
43Ø PRINT AT I+3, J;"███_▐"
44Ø PRINT AT I+4,J;"███▄▟"
45Ø RETURN
5ØØ PRINT AT I,J;"▛▜███"
51Ø PRINT AT I+1,J;"▙▟███"
52Ø PRINT AT I+2,J;"██_██"
53Ø PRINT AT I+3,J;"███▛▜"
54Ø PRINT AT I+4,J;"███▙▟"
55Ø RETURN
6ØØ PRINT AT I,J;"█████"
61Ø PRINT AT I+1,J;"█_█_█"
62Ø PRINT AT I+2,J;"█████"
63Ø PRINT AT I+3,J;"█_█_█"
64Ø PRINT AT I+4,J;"█████"
65Ø RETURN
7ØØ PRINT AT I,J;"▛▜█▛▜"
71Ø PRINT AT I+1,J;"▙▟█▙▟"
72Ø PRINT AT I+2,J;"██_██"
73Ø PRINT AT I+3,J;"▛▜█▛▜"
74Ø PRINT AT I+4,J;"▙▟█▙▟"
75Ø RETURN
8ØØ PRINT AT I,J;"█▀█▀█"
81Ø PRINT AT I+1,J;"█▄█▄█"
```

```
820 PRINT AT I+2,J;"█_█_█"
830 PRINT AT I+3,J;"█▀█▀█"
840 PRINT AT I+4,J;"█▄█▄█"
850 RETURN
```

Points of Interest

Line 30: This generates the die value—an integer between 1 and 6.

Line 60: This line is a calculated **GOSUB** statement which contains a formula that when evaluated sends control to the routine that generates the correct display.

Line 105–280: These lines handle the testing of the throw and determine where to send control, depending upon the rules of the game.

Lines 300–850: These are devoted entirely to generating the sophisticated display.

Possible Adaptations

1. If you are an energetic and enterprising keyboarder, you can try your hand at improving the graphics display. The 6 die looks pretty much like the real thing, you may have noticed, but some of the others—the 1 and 2 dice, for example—show very large white marks to indicate their value. By reworking the graphics symbols in lines 300 and following, you can revamp the display to more closely simulate what real dice look like. This takes careful planning and keyboarding, but you should be able to do it.

2. With the addition of the RAM pack you may wish to add betting routines at line 900. This may be implemented by

changing lines 170 and 290 to read **GOTO** 900. At the end of the program it would then be possible to branch back to line 20.

Chapter 6 BACCARAT

- The Game
- Program 6: BACCARAT
- Points of Interest
- Possible Adaptations

The Game

The rules for this game may vary somewhat from those used in your local casino; for the purposes of this game, the rules are as follows:

- The two cards dealt to each player must add up to 9 or as close to it as possible.
- All cards 10 or above count for 0.
- Any point value greater than 9 is taken as 10 less. Therefore a score of 15 is regarded as 5 and a score of 10 is counted as 0.
- You play against the computer, but you must go first with both cards up. (The computer always has its first card down.)

- Owing to the limitations imposed by the 2K memory, the following conventions have been adopted:

  (1) The player's score is displayed to the left of the screen.

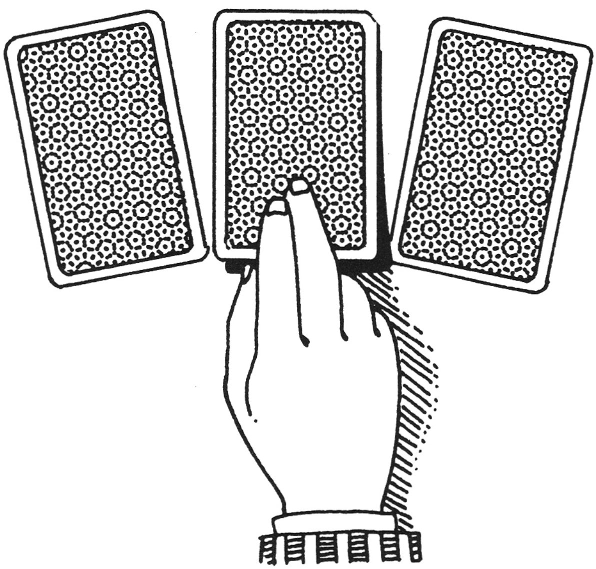
  (2) The computer's score is displayed to the right of the screen.

  (3) Card values are represented by Y (for You) and C (for Computer).

  (4) CD means: Another CarD?

  (5) Y means Yes; N means No. (Hit the ENTER key after either response.)

  (6) After the third card is drawn, the computer will decide if it requires a third card. (The maximum allowed is three.)



Important Note: Due to the extensive use of all memory space in this program, it is recommended that after the program is typed in you type in CLEAR and then ENTER before the program is run. One more note; the permissible card values are A234567890JQK, where the 0 represents 10. Go to it, and good luck!

Program 6: BACCARAT

```
105 LET X$="A234567890JQK"
110 PAUSE 90
112 CLS
115 LET X=1
120 LET Z=1
130 LET P=0
140 FOR T=1 TO 2
170 GOSUB 800
175 GOSUB 900
180 PRINT AT X+1,Z+1;Y$
190 LET P=P+Y
200 LET X=X+3
210 LET Z=Z+2
220 NEXT T
230 GOSUB 1000
260 PRINT AT 19,1;"Y:_";P
300 LET R=1
310 LET S=17
320 LET D=0
```

```
330 FOR T=1 TO 2
340 GOSUB 850
350 GOSUB 900
400 LET D=D+Y
405 IF T=1 THEN GOTO 420
410 PRINT AT R+1, S+1;Y$
420 LET R=R+3
430 LET S=S+2
440 NEXT T
445 GOSUB 1100
450 PRINT AT 19,17;"C:"
460 PRINT AT 21,1;"CD?"
470 INPUT Q$
480 IF Q$="N" THEN GOTO 550
490 GOSUB 800
500 GOSUB 900
510 PRINT AT X+1,Z+1;Y$
515 LET P=P+Y
520 GOSUB 1000
545 PRINT AT 19,4;P
550 PRINT AT 19,20;D
555 IF D=P THEN GOTO 600
560 IF D>P THEN GOTO 650
563 IF T=0 THEN GOTO 675
565 GOSUB 850
570 GOSUB 900
575 PRINT AT R+1,S+1;Y$
```

```
580 GOSUB 1100
590 PRINT AT 19,20;D
596 LET T=0
597 GOTO 555
600 PRINT AT 21,1;"TIE"
610 GOTO 110
650 PRINT AT 21,1;"LOST"
660 GOTO 110
675 PRINT AT 21,1;"WON"
680 GOTO 110
800 LET I=X
810 LET J=Z
820 RETURN
850 LET I=R
860 LET J=S
870 RETURN
900 PRINT AT I,J;"▛▀▀▜"
910 FOR K=I+1 TO I+3
920 PRINT AT K,J;"▌_ _▐"
930 NEXT K
940 PRINT AT I+4,J;"▙▄▄▟"
950 LET Y=INT (RND*13+1)
960 LET Y$=X$(Y)
965 IF Y>10 THEN LET Y=10
970 RETURN
1000 IF P<10 THEN RETURN
1010 LET P=P-10
```

```
1Ø2Ø GOTO 1ØØØ
11ØØ IF D<1Ø THEN RETURN
111Ø LET D=D-1Ø
112Ø GOTO 11ØØ
```

Points of Interest

Lines 800–820 and 850–870: Two subroutines are used to copy the card setup. They are located in lines 800–820 and 850–870. This permits the subroutine in lines 900–970 to be used for both the player's and the computer's cards.

Lines 900–970: The subroutine in these lines creates a card image; it also randomly selects a card value.

Possible Adaptations

1. If more than the computer's regular 2K memory is available, you may want to insert a betting routine at line 1200 and send control there in lines 610, 660, and 680.

2. You might want to ensure that the cards selected are unique. This can be done by including a dimensioned array: M(4 X 13). In such a setup, all the elements are initialized to zero, and the card selection is modified to include the selection of a suit (spades, hearts, clubs, or diamonds) using a random selection of 1, 2, 3 or 4.

3. After choosing a random number 1 to 13 and another 1 to 4 (say, 10 and 2), a check should be made of the value of M(2,10). If it's equal to 0, the card has not yet been used and is therefore available. Once it is used, however, that element should be set to 1. If M(2,10) is equal to 1, the card has already been used and two new values should be generated.

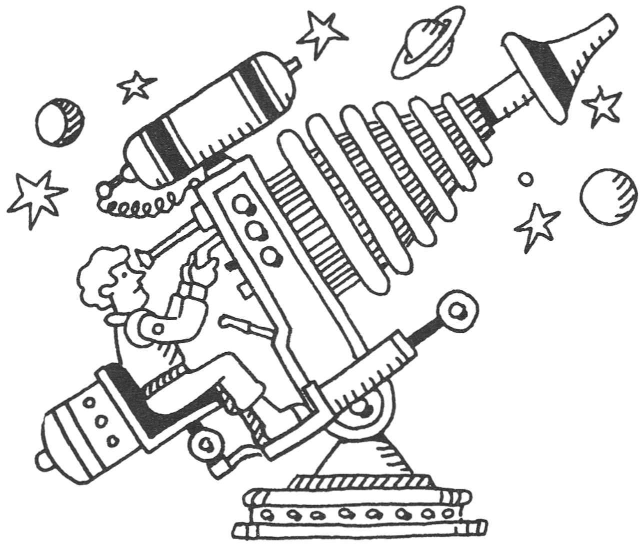
Chapter 7 SPACE PROBE

- The Game
- Program 7: SPACE PROBE
- Points of Interest
- Possible Adaptations

The Game

In this game you are playing the role of a fleet commander on a star base under attack by *invisible* aliens. You may fire a probe in one of four directions using one of the four "arrow commands" located on keys 5, 6, 7 and 8. Your shot is displayed by asterisks. If you are successful and actually hit an alien invader, this will be indicated on your radar panel, on the lower portion of the screen. You will be awarded 100 points for each hit. If you miss, however, your radar panel will register the miss. This has the effect of damaging your probe launcher, causing the distance it can fire to be shortened. To add to the problems, any damage sustained by your launcher is irreparable.

The game terminates when the range of the probe becomes so short as to be ineffective. As soon as the point is reached that your shots can no longer fire far enough, the message GAME OVER is displayed.



Program 7: SPACE PROBE

```
 5Ø LET A=Ø
 6Ø LET B=2Ø
 7Ø LET C=3Ø
 8Ø LET P=Ø
1ØØ PAUSE 3Ø
1Ø3 CLS
1Ø5 PRINT AT 1Ø,15;"▒"
11Ø PRINT AT 11,14;"▒█▒"
12Ø PRINT AT 12,15;"▒"
13Ø LET X=INT (RND*3+5)
135 LET Z=Ø
14Ø LET F=Ø
```

```
200 IF INKEY$<>"" THEN GOSUB 45Ø
21Ø IF F=1 THEN GOTO 3ØØ
215 IF Z=Ø THEN GOTO 13Ø
22Ø LET A=A+1
23Ø LET B=B-.5
24Ø LET C=C-1
25Ø PRINT AT 21,12;"MISSED"
255 IF A=11 OR B<13 THEN GOTO 9ØØ
26Ø GOTO 1ØØ
3ØØ LET P=P+1ØØ
31Ø PRINT AT 21,12;"A_HIT"
32Ø PRINT AT 21,20;"PTS:_";P
33Ø GOTO 1ØØ
45Ø LET Z=1
455 IF INKEY$="5" THEN GOTO 5ØØ
46Ø IF INKEY$="6" THEN GOTO 6ØØ
47Ø IF INKEY$="7" THEN GOTO 7ØØ
48Ø IF INKEY$="8" THEN GOTO 8ØØ
49Ø RETURN
5ØØ FOR I=13 TO A STEP -1
51Ø PRINT AT 11,I;"*"
53Ø NEXT I
54Ø IF X=5 THEN LET F=1
59Ø RETURN
6ØØ FOR I=13 TO B
61Ø PRINT AT I,15;"*"
63Ø NEXT I
```

Be sure to include the decimal point in .5. (← line 23Ø)

```
640 IF X=6 THEN LET F=1
690 RETURN
700 FOR I=9 TO A STEP -1
710 PRINT AT I,15;"*"
730 NEXT I
740 IF X=7 THEN LET F=1
790 RETURN
800 FOR I=17 TO C
810 PRINT AT 11,I;"*"
830 NEXT I
840 IF X=8 THEN LET F=1
890 RETURN
900 PRINT AT 21,10;"GAME_OVER"
```

Points of Interest

Lines 105–120: These set up the star base.

Line 120: This line determines if an alien is present.

Lines 220–240: The range of the space probe is reduced by these lines.

Lines 455–490: The direction in which the probe is to be launched—up, down, left, or right—is determined by these lines.

Lines 500–590: They set up the probe to be fired left.

Lines 600–690: Launch the probe downwards.

Lines 700–790: Launch the probe upwards.

Lines 800–890: Launch the probe to the right.

Possible Adaptations

If some additional memory space is available, each alien invader can be made to orbit the station. Each time a miss is registered, some action can be initiated, such as bringing the alien closer to the station by one unit. For each hit, the alien can be repelled one unit further from the base.

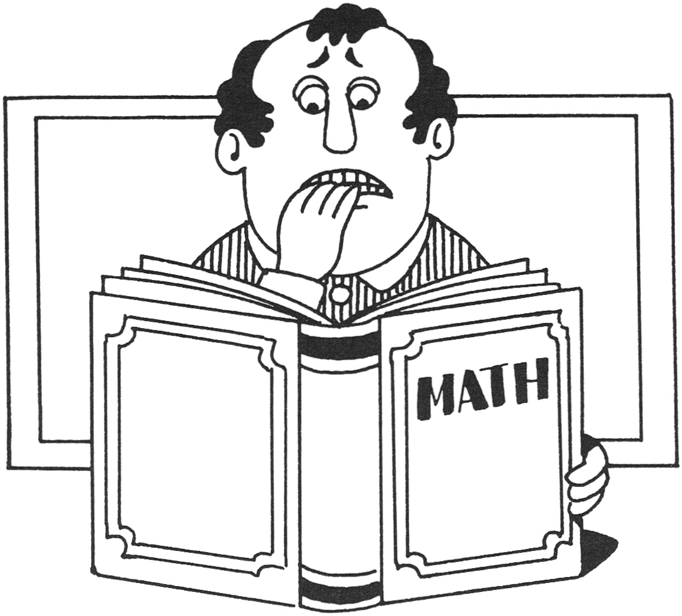
Chapter 8 MATH QUIZ

- The Game
- Program 8: MATH QUIZ
- Points of Interest
- Possible Adaptations

The Game

In this game the computer will test your skills of elementary arithmetic. You really don't have to get nervous because all the program does is to exercise your ability to add, subtract, multiply, and divide. Which one of these is selected is up to the computer, not you. All you have to do is to answer the questions correctly!

Each time you play this game, the values selected for calculation will be different because they are randomly selected by the computer. When the question is displayed, you are invited to select the correct answer from four possible answers—a sort of multiple-choice quiz, if you will. The choices have been arranged so that each of the answers could easily be mistaken for the correct one, so you really have to be alert to do well on this quiz. There is no time limit imposed on the user, so you can take your time before making your choice.



Each time your selection is correct, the score is incremented by 1. Each time you select a wrong result, you are notified and you are given the correct result. To terminate the program, simply type in 0 for an answer. The best of luck to you!

Program 8: MATH QUIZ

```
1ØØ LET C=Ø
1Ø5 LET P=Ø
11Ø DIM Q(4)
12Ø LET K=INT (RND*4+1)
125 LET P=P+1
13Ø GOSUB K*1ØØ+2ØØ
```

```
135 PRINT AT 3,12;"MATH QUIZ"
14Ø LET S=INT (RND*4+1)
15Ø LET Q(S)=A
16Ø GOSUB 7ØØ
17Ø PRINT AT 8,4;"QUESTION_==>_";X;Y$;Y;
    "_=_?"
18Ø PRINT AT 11,1;
19Ø FOR J=1 TO 4
2ØØ PRINT J;"._";Q(J);"__";
21Ø NEXT J
22Ø PRINT AT 17,8;"PICK_1,2,3_OR_4"
23Ø INPUT G
24Ø IF G=Ø THEN GOTO 9ØØ
25Ø IF Q(G)=A THEN GOTO 8ØØ
26Ø PRINT AT 17,8;"WRONG,_IT_WAS_NO._";
    S
27Ø GOTO 82Ø
3ØØ LET X=INT (RND*9Ø+1Ø)
31Ø LET Y=INT (RND*9Ø+1Ø)
32Ø LET A=X+Y
33Ø LET Y$="_+_"
34Ø RETURN
4ØØ LET X=INT (RND*9Ø+1ØØ)
41Ø LET Y=INT (RND*9Ø+1Ø)
42Ø LET A=X-Y
43Ø LET Y$="_-_"
44Ø RETURN
```

Watch that decimal point in the first literal. (→ line 2ØØ)

```
5ØØ LET X=INT (RND*2Ø+4)
51Ø LET Y=INT (RND*2Ø+4)
52Ø LET A=X*Y
53Ø LET Y$="_*_"
54Ø RETURN
6ØØ LET Y=INT (RND*1Ø+2)
61Ø LET X=INT (RND*2Ø+1Ø)*Y
62Ø LET A=X/Y
63Ø LET Y$="_/_"
64Ø RETURN
7ØØ FOR I=1 TO 4
71Ø IF Q(I)=A THEN GOTO 76Ø
72Ø LET Q(I)=INT (RND*(I+1)-3)*5+A
73Ø FOR Z=1 TO (I-1)
74Ø IF Q(Z)=Q(I) THEN GOTO 72Ø
75Ø NEXT Z
755 IF Q(I)=A THEN GOTO 72Ø
76Ø NEXT I
77Ø RETURN
8ØØ LET C=C+1
81Ø PRINT AT 17,8;"***_CORRECT_***"
82Ø PRINT AT 21,1Ø;"SCORE:_";C;"/";P
83Ø PAUSE 1ØØ
84Ø CLS
85Ø GOTO 12Ø
9ØØ LET D=C/(P-1)*1ØØ
91Ø CLS
```

Don't forget line 920, at the top of the next page.

```
920 PRINT AT 11,8;"FINAL_SCORE:_";INT D
```

Points of Interest

Line 130: A calculated **GOSUB** brings you to the appropriate subroutine for a question on addition, subtraction, multiplication, or division.

Line 170: The question with the randomly selected operation stored in Y$ is printed out by this line.

Lines 300–640: These lines generate the random numbers to be used in the questions.

Lines 700-770: These represent the routine that generates the phony alternatives, all of which correspond in some degree to the correct answer.

Lines 730–755: These lines ensure that each choice is unique. In other words it avoids the possibility of duplication of alternatives. Line 730 checks whether a particular incorrect answer has been used yet. Line 755 makes sure that an incorrect answer is not equal to the correct answer.

Lines 800–820: These lines are activated when the answer is correct; they are responsible for displaying the score.

Lines 900–920: If the user types in 0 for an answer, these lines are reached. This has the effect of calculating the final score, clearing the screen, printing out the final score, and terminating the program.

Possible Adaptations

It would make for an even more challenging game if the responses not only had to be correct, but also had to be entered within a limited period of time. This could be implemented in lines 220–240 using the familiar **PAUSE** and **INKEY$** statements.

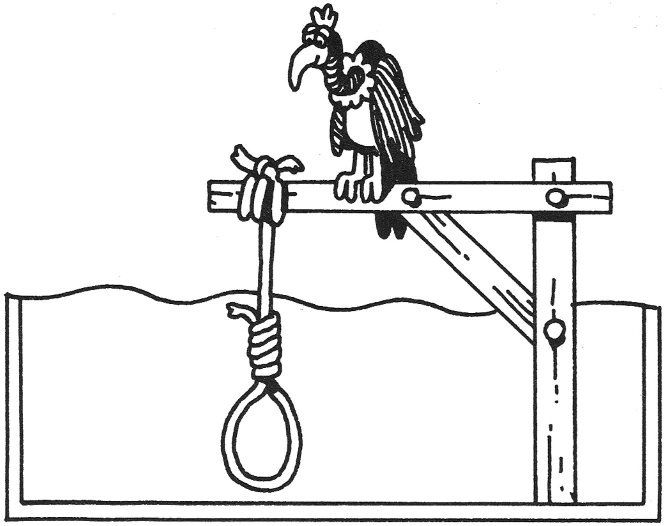
Chapter 9 HANG 'EM

- The Game
- Program 9: HANG'EM
- Points of Interest
- Possible Adaptations

The Game

Who among us has not been fascinated from early childhood by the game of Hangman? With the aid of your T/S 1000, it is possible to play this game electronically! This version of Hangman responds to each incorrect answer by graphically building up the scaffold and the body right there on the screen.

The rules are that the hidden words may be of varying length. When a letter is typed in, the hidden word is scanned and the letter, if present, is entered in the display in the position where it belongs. If a particular letter appears more than once, it is inserted in all the appropriate positions simultaneously. If the selected letter is not correct, then another part of the scaffold is completed and is displayed on the right of the screen. The seventh incorrect guess will result in the termination of the game.



Program 9: HANG'EM

```
1Ø LET X$="IMPOSSIBLEPROGRAM___
   YESTERDAY_RIDICULOUSGRAPHICS__
   TELEPHONE_UPWARD_____CUSTOMER
   __CLARIFY___"
2Ø LET Y=INT (RND*8+1)
3Ø LET W$=X$((Y-1)*1Ø+1 TO (Y-1)*1Ø+
   1Ø)
4Ø LET H$=W$
5Ø FOR I=1 TO 1Ø
6Ø IF W$(I)<>"_" THEN GOTO 9Ø
7Ø GOTO 1ØØ
9Ø NEXT I
```

```
100 LET L=I-1
110 LET D$="▒▄▒▄▒▄▒▄▒▄▒▄" (1 TO L)
130 PRINT AT 2,12;"HANG_EM_HIGH"
140 LET E=Ø
170 PRINT AT 7,9;"GUESS_A_LETTER"
180 PRINT AT 17,3;D$
190 INPUT L$
191 LET F=Ø
220 FOR I=1 TO L
230 IF L$=W$(I) THEN GOTO 300
240 NEXT I
250 IF F=1 THEN GOTO 170
260 LET E=E+1
270 GOTO E*100+500
300 LET D$(I)=W$(I)
310 LET W$(I)="_"
320 LET F=1
330 IF D$=H$(1 TO L) THEN GOTO 410
340 GOTO 220
410 PRINT AT 21,16;"YOU GOT IT"
420 PRINT AT 17,3;D$
430 STOP
600 PRINT AT 10,25;"▛▀▀▀▀▜█"
610 FOR Q=11 TO 19
620 PRINT AT Q,31;"█"
630 NEXT Q
640 GOTO 170
```

```
700 PRINT AT 11,24;"▛▜"
710 PRINT AT 12,24;"▙▟"
720 GOTO 170
800 PRINT AT 13,22;"▄▄▟"
810 GOTO 170
900 PRINT AT 13,25;"▙▄▄"
910 GOTO 170
1000 PRINT AT 14,24;"██"
1010 PRINT AT 15,24;"██"
1020 GOTO 170
1100 PRINT AT 16,24;"▌▐"
1110 PRINT AT 17,23;"▐_ _▌"
1120 GOTO 170
1200 PRINT AT 18,23;"▟_ _▙"
1210 PRINT AT 20,23;"HUNG"
```

Points of Interest

Line 10: All the words are stored in the string variable X$ in line 10. It can be seen that they are of varying length. However, to insure that the words are correctly accessed, each word of less than ten letters is padded out to the right with blanks.

Line 30: In this line a word is selected and stored into W$.

Line 40: Stores a copy of W$ in H$.

Lines 50–100: These lines determine the actual length of the word selected by searching until the first space is encountered and by setting L equal to the calculated length.

Line 110: This line sets up the space in the display for the hidden letters.

Lines 220–330: The dashes symbolizing the unknown letters are replaced by the actual letters if they are indeed present in the hidden word.

Line 270: This is a calculated **GOTO** statement that sends control to the part of the program that sets up the appropriate display.

Line 600: This line and all the lines to the end of the program set up the entire display for the hanged man.

Important Notes: For this program to run you must first hit the CLEAR and ENTER keys. Also, the list of unknown words is limited to eight.

Possible Adaptations

1. You should feel free to change the 8 words in line 10 to any words of your own choosing. However, be sure that they are always padded out with blanks to the right, thereby creating uniform ten-letter words.

2. If extra memory is available, you may be tempted to expand the pool of hidden words. Should you do this, be sure to change line 20 to reflect the new number of words. If you do not do this, the extra words will reside in X$ but will never be chosen.

3. Another and much more challenging modification is to allow for guessing the entire word without having to type in the individual letters. Can you figure out how to do this?

4. The game could probably be improved if the letters already selected were printed out for you on the lower part of the screen. This would be particularly useful for the younger folks. Try it.

5. In addition, it would be very nice if some error-trapping mechanism were built into the program to avoid the possiblity of selecting a letter which has already been selected.

6. If you are a grammarian, you know that "HUNG," as it appears in line 1210, should really be "HANGED." Change it if you like.

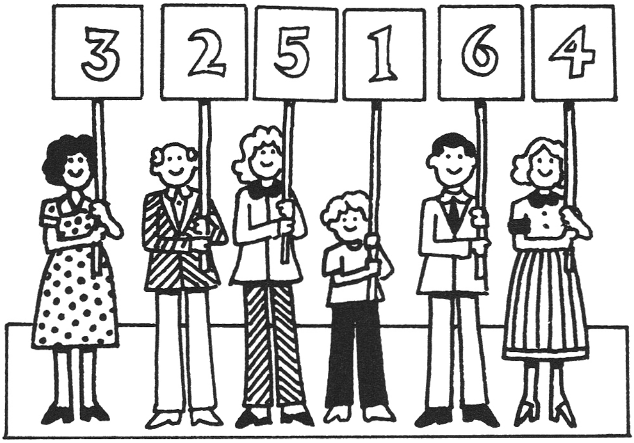
Chapter 10 SEQUENCE?

- The Game
- Program 10: SEQUENCE?
- Points of Interest
- Possible Adaptations

The Game

This program is a variation of the electronic game called "Simon." It allows one to four players to participate, and with each round the difficulty is increased. You are asked to repeat a sequence of numbers which is flashed in the middle of the screen, one number at a time. The game starts with 4 such digits. Each player is asked to repeat a random sequence of numbers. Any player who fails is immediately eliminated from the game, and the correct sequence is displayed.

The next round includes all the remaining players, but the number of digits displayed is increased by one. Moreover, the flashing of the numbers is speeded up. As you will find out when you run the program, the numbers are flashed very fast indeed. The maximum sequence possible contains 10 numbers, but it is safe to bet that you will never reach it!



Once all the players are eliminated, the score for each is listed. If you enjoy a stiff challenge—this is it!

Program 10: SEQUENCE?

```
1ØØ DIM P(4)
1Ø5 DIM S(4)
11Ø DIM H$(1Ø)
12Ø LET X$="Ø123456789"
13Ø LET N=4
14Ø FOR I=1 TO 4
145 CLS
15Ø IF P(I)=1 THEN GOTO 27Ø
```

```
16Ø PRINT AT 3,10;"PLAYER_";I;"_GOES"
163 PRINT AT 5,12;N;"_NUMBERS"
165 PAUSE 1ØØ
17Ø FOR J=1 TO N
175 LET K=INT (RND*1Ø+1)
18Ø LET H$(J)=X$(K)
19Ø PRINT AT 11,15;H$(J)
2ØØ FOR H=1 TO (1Ø-N)
2Ø5 NEXT H
21Ø PRINT AT 11,15;"_"
22Ø NEXT J
23Ø PRINT AT 16,5;"REPEAT_THE_SEQUENCE
    _NO._";I
24Ø INPUT A$
25Ø IF A$=H$(1 TO N) THEN GOSUB 4ØØ
26Ø IF A$<>H$(1 TO N) THEN GOSUB 5ØØ
265 PAUSE 15Ø
27Ø NEXT I
28Ø IF N<1Ø THEN LET N=N+1
29Ø FOR T=1 TO 4
3ØØ IF P(T)=Ø THEN GOTO 14Ø
31Ø NEXT T
32Ø CLS
33Ø PRINT AT 3,6;"ALL_PLAYERS_ARE_OUT"
34Ø PRINT AT 7,12;"SCORES"
35Ø FOR T=1Ø TO 13
36Ø PRINT AT T,7;"PLAYER_";T-9;"_===>_";
    S(T-9)
```

```
370 NEXT T
380 STOP
400 PRINT AT 18,10;"OKAY,_NO._";I
410 LET S(I)=S(I)+1
420 RETURN
500 PRINT AT 18,10;"SORRY_NO._";I
510 LET P(I)=1
520 PRINT AT 11,2;"PLAYER_";I;"_IS_OUT_
    OF_THE_GAME"
530 PRINT AT 13,10;"IT_WAS_";H$(1 TO N)
540 RETURN
```

Points of Interest

Line 120: The ten digits 0–9 are stored in the string variable X$. This is for ease of extracting a digit.

Lines 190–210: In these lines the digits are displayed and the time interval is varied.

Lines 250–260: In these lines a test is made to determine if the typed-in sequence is correct.

Lines 320–380: These clear the screen and display the final scores.

Lines 400–420: These confirm a correct response and add 1 to the score of each successful player. If fewer than 4 players are desired for a game, those not being played may be eliminated by typing in an incorrect response immediately following the first round.

Possible Adaptations

1. The input may be timed by inserting appropriate instructions in lines 230–250

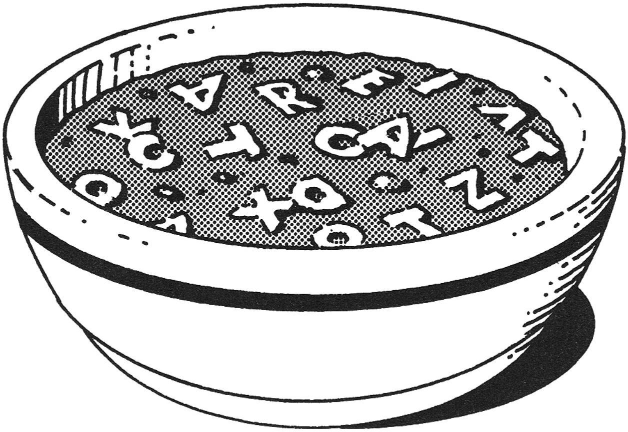
2. If the pace of the game is too fast, lines 200–205 (which represent a simple delay loop) may be adjusted. For faster play, the number should be reduced; for slower play, it should be increased.

Chapter 11 SCRAMBLE

- The Game
- Program 11: SCRAMBLE
- Points of Interest
- Possible Adaptations

The Game

If you are fond of the game "Jumbles," this game is for you. The computer has stored in its memory 10 different words, each of which is six characters in length. The computer randomly selects one of them, scrambles it up, and displays it on the screen for a period of 3 to 4 seconds. You are required to decipher the scrambled word and type it in before the time limit is exceeded. If you succeed, you gain a point. If you either miss the deadline or type in an incorrect answer, you get 0 for your efforts. At the end of each round the computer asks you if you want to go around again. It is only fair to say that some of the words are very difficult to decipher, especially for the uninitiated.



Program 11: SCRAMBLE

```
100 LET X$="MELLOWFASTERGOPHERJACKA
    LRETURNVICTORDELETESCROLLRANDO
    MJUMBLE"
105 LET T=0
106 LET C=0
110 DIM A(6)
115 DIM D$(6)
120 LET Y=INT (RND*10+1)
125 LET T=T+1
130 LET H$=X$((Y-1)*6+1 TO (Y-1)*6+6)
135 GOSUB 400
140 FOR I=1 TO 6
```

```
15Ø LET J=INT (RND *6+1)
16Ø IF A(J)<>Ø THEN GOTO 15Ø
17Ø LET A(J)=1
18Ø LET D$(I)=H$(J)
19Ø NEXT I
193 IF D$=H$ THEN GOTO 135
195 CLS
2ØØ PRINT AT 2,8;"SCRAMBLED_WORDS"
21Ø PRINT AT 8,7;"WHAT_IS_THIS_WORD?"
22Ø PRINT AT 12,13;D$
225 PRINT AT 13,13;"▄▄▄▄▄▄"
23Ø PAUSE 2ØØ
24Ø IF INKEY$<>"" THEN GOTO 3ØØ
25Ø PRINT AT 21,12;"TIME_IS_UP"
26Ø GOTO 325
3ØØ INPUT R$
31Ø IF R$=H$ THEN GOTO 32Ø
315 PRINT AT 21,13;"WRONG"
317 GOTO 325
32Ø PRINT AT 21,12;"CORRECT"
323 LET C=C+1
325 PRINT AT 21,25;C;"/";T
33Ø PAUSE 1ØØ
34Ø PRINT AT 21,1Ø;"MORE?_Y_OR_N"
35Ø INPUT Q$
36Ø IF Q$="Y" THEN GOTO 12Ø
37Ø STOP
```

```
400 FOR E=1 TO 6
410 LET A(E)=0
420 NEXT E
430 RETURN
```

Points of Interest

Lines 140–190: The loop in lines 140–190 scrambles the letters in the selected word.

Line 193: This line ensures that the scrambled word is not the same as the unscrambled word.

Lines 220–225: Take care of displaying of scrambled letters.

Lines 230–240: The timing mechanism is lodged in lines 230–240.

Lines 250–330: These test the user's response and display the results.

Line 340: After each round the user is asked if he or she wishes to go around again. A response of Y for yes sends control to line 120, where another word is selected.

Possible Adaptations

1. The list of ten words may be changed at the discretion of the user. The words must, however, have exactly six letters.

2. For increased difficulty, the lengths of the words may be increased. If this is done, lines 110, 115, 130, 140, 150, and 400 must be amended accordingly.

3. The number of words must be reflected in line 120.

4. The time allotted for each word is controlled by the **PAUSE** in line 230. This may be changed to suit your convenience.

Chapter 12 CONCENTRATION

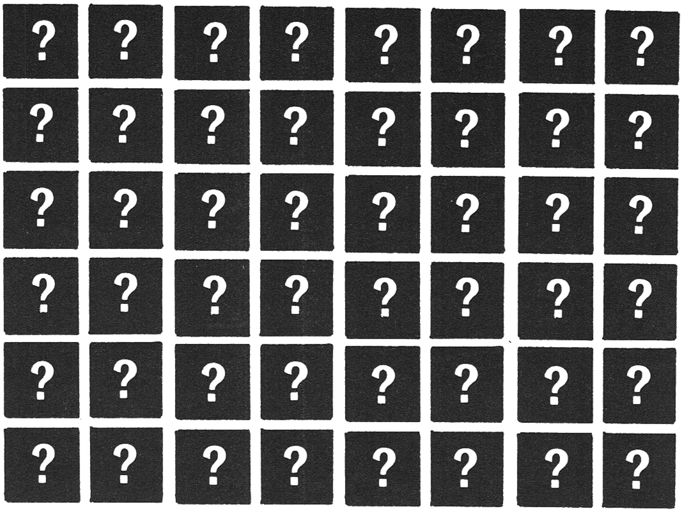
- The Game
- Program 12: CONCENTRATION
- Points of Interest
- Possible Adaptations

The Game

"Concentration" is the two-player computer version of the well-known card game. This version has only 48 electronic "cards," represented by the letters A–L, with 4 occurrences of each letter. The computer randomly shuffles the cards and player number 1 goes first.

The forty-eight cards are represented by a 6 X 8 matrix composed of 6 rows and 8 columns. A selection of a card is made by inputting the *row* number first (1 to 6) and ENTER and then the column number (1 to 8) and ENTER. At this point, the first card selected is revealed and the player is asked to make the second selection. Now the second is revealed. If they match, the player gains a point and continues, with cards being replaced by blanks that can never be accessed again. If the cards do not agree, they are covered up and the second player goes.

The process is repeated for the second player. The game



is terminated when all twenty-four pairs of cards have been matched. The winner is the player who has matched the most pairs of cards.

Important Note: Prior to running this program, hit the CLEAR and ENTER keys. Due to memory restrictions, no error checking is possible in this program. To avoid problems take care to select cards that have not already been matched and removed from the screen.

Another Important Note: Patience is required after the command **RUN** is typed in and the ENTER key is hit. It takes the computer about 60 seconds to generate a great many random numbers, to hide the cards, and to prepare the display.

Program 12: CONCENTRATION

```
1Ø DIM A(6,8)
2Ø LET P1=Ø
```

```
 3Ø LET P2=Ø
 4Ø LET X$="ABCDEFGHIJKL"
 5Ø FOR I=1 TO 12
 6Ø FOR J=1 TO 4
 7Ø LET X=INT (RND*6+1)
 8Ø LET Y=INT (RND*8+1)
 9Ø IF A(X,Y)<>Ø THEN GOTO 7Ø
1ØØ LET A(X,Y)=I
11Ø NEXT J
12Ø NEXT I
13Ø PRINT AT 1,11;"CONCENTRATION"
14Ø PRINT AT 3,1Ø;"1_2_3_4_5_6_7_8"
15Ø FOR I=1 TO 6
16Ø PRINT AT 2*I+3,8;I;
17Ø PRINT "_■_■_■_■_■_■
    _■_■"
18Ø NEXT I
2ØØ LET Z=1
23Ø PRINT AT 17,13;Z;"_GOES"
28Ø GOSUB 3ØØ
29Ø GOTO 23Ø
3ØØ PRINT AT 19,15;"1ST"
31Ø INPUT X
32Ø INPUT Y
41Ø LET F=A(X,Y)
42Ø PRINT AT 2*X+3,2*Y+8;X$(F)
43Ø PRINT AT 19,15;"2ND"
```

```
44Ø INPUT R
45Ø INPUT T
46Ø LET S=A(R,T)
47Ø PRINT AT 2*R+3,2*T+8;X$(S)
475 PAUSE 1ØØ
48Ø IF F=S THEN GOTO 55Ø
482 IF Z=1 THEN GOTO 486
483 LET Z=1
484 GOTO 49Ø
486 LET Z=2
49Ø LET Y$="■"
5ØØ PRINT AT 19,15;"NO_"
51Ø PRINT AT 2*X+3,2*Y+8;Y$
52Ø PRINT AT 2*R+3,2*T+8;Y$
53Ø RETURN
55Ø PRINT AT 19,15;"OK_"
555 LET Y$="_"
56Ø IF Z=1 THEN GOTO 6ØØ
57Ø LET P2=P2+1
58Ø PRINT AT 11,29;P2
59Ø GOTO 62Ø
6ØØ LET P1=P1+1
61Ø PRINT AT 11,3;P1
62Ø IF P1+P2<>24 THEN GOTO 51Ø
```

Points of Interest

Lines 50–120: The nest of loops in lines 50–120 hides the cards in various locations, ensuring that only four occurrences of one card will be placed in memory.

Lines 130–180: The screen is set up for the game in these lines.

Lines 230–320: These request either player to key in the position of the first card.

Lines 410–420: The first card is flipped over in lines 410–420.

Lines 430–450: These allow the player to key in the second card.

Lines 460–470: These flip the second card.

Lines 480–530: The matching of the cards is done in lines 480–486. If no match is made, the card is flipped back in lines 490–530.

Lines 500–530: These lines, in conjunction with line 555, blank out the cards in the event that they match.

Lines 560–620: These lines print out the points scored in the game. The score on the left is for player number 1, and the score on the right is for player number 2.

Possible Adaptations

In order to carry out the modifications that follow it is essential to have more than the 2K of memory provided with the computer.

Some error trapping can be made by arranging in lines 550–555 for a 0 to be inserted in matrix A each time a pair is found. This, in conjunction with additional lines at 300 and 430, could reject the choice of a card no longer available. This could be accomplished in the following way: When inputting a card in lines 320–410, check to see if A(X,Y)=0.

If so, **GOTO** 300 and request the player to enter a new pair of co-ordinates. Also, check if in lines 450–460 A(R,T) = 0. If so, **GOTO** 430 in the same way.

Chapter 13 ALIEN GUNNER

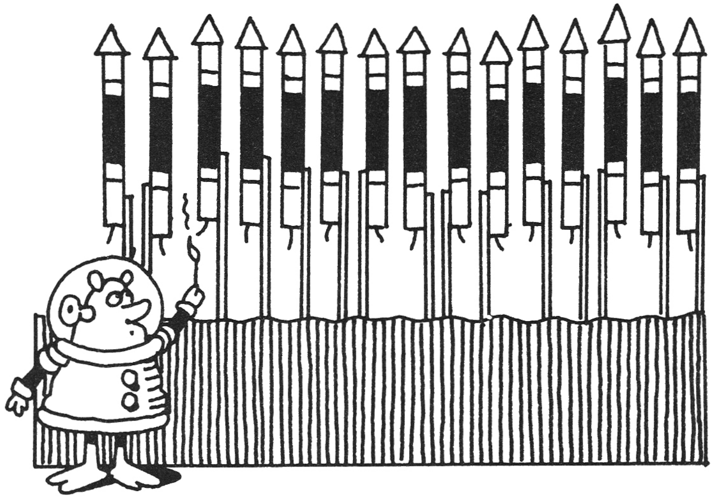
- The Game
- Program 13: ALIEN GUNNER
- Points of Interest
- Possible Adaptations

The Game

This game is called "Alien Gunner." In this scenario you are the last Earth defense base still operational, and you are feverishly warding off hordes of alien attackers swarming in at all levels of the atmosphere.

In order to eliminate aliens you must score a direct hit (no nipping). You are limited to a total of 15 shots. Your available reserve energy is shown on the lower left of the screen. After each shot is fired one unit of reserve energy is removed from the display. If you score a direct hit you will be rewarded by the spectacle of the alien ship disintegrating. Points are awarded according to the level of the attacking alien ship.

In order to fire at an alien located directly above simply press any key. The game is over when your reserves have been reduced to zero.



Program 13: ALIEN GUNNER

```
 1Ø FOR I=Ø TO 28 STEP 2
 2Ø PLOT I,Ø
 3Ø NEXT I
 4Ø LET K=Ø
 45 LET W=Ø
 5Ø FOR Z=Ø TO 14
 55 IF W<>Ø THEN GOTO 9Ø
 6Ø LET F=INT (RND*4+1)
 7Ø LET S=INT (RND*29+2)
 8Ø PRINT AT F,S-1;"▄X▄"
 9Ø FOR I=Ø TO 63
1ØØ PLOT I,2
```

```
11Ø UNPLOT I,2
12Ø IF INKEY$<>"" THEN GOTO 15Ø
13Ø NEXT I
14Ø GOTO 9Ø
15Ø FOR J=2 TO 41
16Ø PLOT I,J
17Ø UNPLOT I,J
18Ø NEXT J
19Ø UNPLOT 28-2*Z,Ø
2ØØ IF I=2*S OR I=2*S+1 THEN GOTO 22Ø
2Ø5 LET W=1
21Ø GOTO 28Ø
22Ø GOSUB (F+2)*1ØØ
225 LET W=Ø
23Ø PRINT AT F,S-1;"▄▒▄"
24Ø PAUSE 5Ø
25Ø PRINT AT F,S-1;"___"
26Ø PAUSE 5Ø
27Ø PRINT AT 21,27;K
28Ø NEXT Z
29Ø GOTO 7ØØ
3ØØ LET K=K+1ØØØ
35Ø RETURN
4ØØ LET K=K+5ØØ
45Ø RETURN
5ØØ LET K=K+25Ø
```

```
55Ø RETURN
6ØØ LET K=K+1ØØ
61Ø RETURN
7ØØ PRINT AT 1Ø,12;"GAME OVER"
```

Points of Interest

Lines 60–80: The point at which an alien ship appears on the screen is determined in lines 60–80.

Lines 90–140: These lines not only control the radar blip that continuously scans the screen; they also handle the read from the keyboard.

Lines 150–180: These lines launch the shot in a vertical direction.

Line 190: This line uses a formula to remove one unit of supplies from the reserves after each shot is fired.

Line 200: It is executed if a direct hit is made. The line directs control to the routine that adjusts the score.

Lines 230–260: These lines control the display of the ship in two stages. The first is when it is in the process of disintegrating and the second when it is in the stage of final destruction.

Lines 300–610: Point values are added to the score by the routine in lines 300–610 in conjunction with line 220.

Possible Adaptations

1. You might be tempted to increase the number of shots from 15 to any number of your choice.

2. If you have artistic inclinations you might want to redesign the alien ships. This would also be a good practice in working the graphics mode.

Chapter 14 BARRIERS

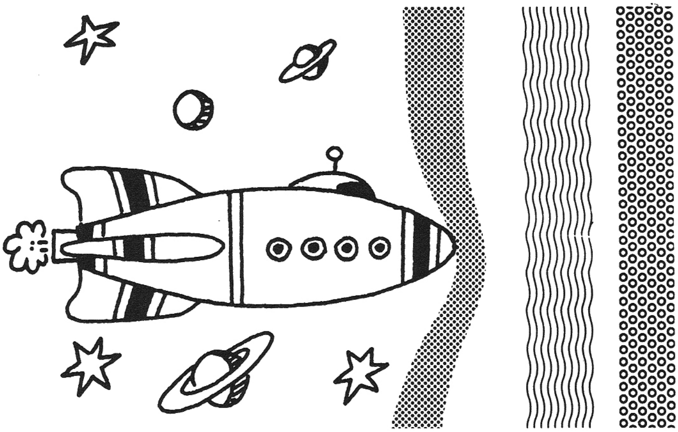
- The Game
- Program 14: BARRIERS
- Points of Interest
- Possible Adaptations

The Game

While you are in a combative spirit, how about tackling this game? Here you are a commander of a small scouting spacecraft in desperate need of supplies. The nearest base, however, is located on the other side of the formidable and treacherous Forbidden Zone. In order to reach the base you must cross three electronic force fields. Each field, however, has weak points. The idea of the game is to reach the Forbidden Zone by slipping through the chinks.

You may hit any key to advance your ship, which always starts out at the bottom left of the screen. You cannot linger in one of the gaps, nor can you hit one of the force fields, unless you want your family and friends to be sent appropriate condolences.

If you are gifted with uncanny skill and somehow manage to find your way safely through all three of the force fields, you may then pick up supplies and receive your



points. If you hit the outer rim of space (the edge of the screen) your spacecraft is mysteriously teleported back to the initial position and the flight recommences. The game terminates the moment you crash into one of the three fields or linger too long in any one of the gaps.

Program 14: BARRIERS

```
4 PRINT AT 2,8;"THE_FORBIDDEN_ZONE"
5 LET X$="█████████_████████
  █_"
6 LET Z$="__████████__██████
  ██"
7 LET B$="█████___████___██
  ___"
8 LET W=Ø
1Ø LET I=2Ø
15 LET P=Ø
```

```
 20 LET J=14
 30 PRINT AT I,J;"A"
 40 FOR K=1 TO 100
 50 LET Y$=X$(1)
 60 LET X$(1 TO 19)=X$(2 TO 20)
 70 LET X$(20)=Y$
 80 LET T=10
 90 GOSUB 300
100 PRINT AT T,6;X$
110 IF INKEY$<>""THEN GOSUB 400
115 IF P=1 THEN GOTO 10
120 LET Y$=Z$(20)
130 LET Z$(2 TO 20)=Z$(1 TO 19)
140 LET Z$(1)=Y$
150 LET T=12
160 GOSUB 320
170 PRINT AT T,6;Z$
180 IF INKEY$<>"" THEN GOSUB 430
190 LET Y$=B$(1)
200 LET B$(1 TO 19)=B$(2 TO 20)
210 LET B$(20)=Y$
220 LET T=14
230 GOSUB 340
240 PRINT AT T,6;B$
250 IF INKEY$<>"" THEN GOSUB 460
260 NEXT K
270 GOTO 40
```

```
300 IF I=1Ø AND X$(1Ø)="■" THEN GOTO 5ØØ
310 RETURN
320 IF I=12 AND Z$(1Ø)="■" THEN GOTO 5ØØ
330 RETURN
340 IF I=14 AND B$(1Ø)="■" THEN GOTO 5ØØ
350 RETURN
400 IF I-1=1Ø AND X$(1Ø)="■" THEN GOTO
    5ØØ
405 IF I>=8 THEN GOTO 47Ø
408 PRINT AT I,J;"_"
410 LET W=W+5ØØ
415 LET P=1
420 PRINT AT 21,1;W
425 RETURN
430 IF I-1=12 AND Z$(1Ø)="■" THEN GOTO
    5ØØ
440 GOTO 47Ø
460 IF I-1=14 AND B$(1Ø)="■" THEN GOTO
    5ØØ
470 PRINT AT I,J;"_"
475 LET I=I-1
480 PRINT AT I,J;"A"
490 RETURN
500 PRINT AT I,J;"_"
510 PRINT AT T,J;"A"
520 PRINT AT 2Ø,1Ø;"YOU CRASHED"
```

Points of Interest

Lines 5–7: The force field barriers are set up in these lines.

Lines 40–270: These control the rightward movement of the top and lower barriers and the leftward movement of the middle barrier. (At the same time, the lines are constantly checking on the movement of the spacecraft.)

Lines 300–350: These check to see whether you are lingering too long in a gap.

Lines 400–490: These check to see if your spacecraft is about to crash.

Lines 500–520: These change the letter A to its inverse-video form to represent a crash.

Possible Adaptations

1. You might want to add one or more barriers. Be aware, however, that this will slow down the game.
2. An extremely difficult but satisfying modification you might want to consider is to control the ship in three directions (up, left, and right).
3. Another challenging change to make is to give your spacecraft continuous motion in the last direction typed. (In other words, once you type in the left key, your spacecraft will continue to move left until you hit right.) Since you will need to stop, a key to accomplish this action must be added to the program.

Chapter 15 FILL 'ER UP

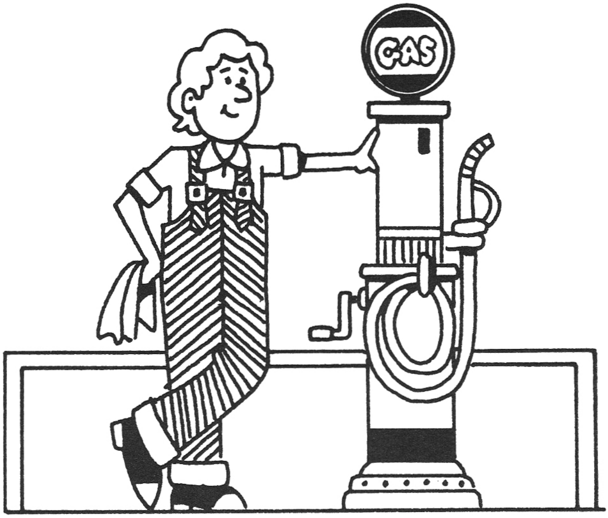
- The Game
- Program 15: FILL 'ER UP
- Points of Interest
- Possible Adaptations

The Game

Regardless of how well you have fared up until now, the next game is very likely to trip you up. If you are the kind of person who never closes his eyes to a severe challenge, start keying in the next program; but don't say you were not warned!

The purpose of the game is to fill in the large, black box by controlling the passage of the light, shaded box. The large box must be filled in completely. In order to control the direction of the light box, use the 5, 6, 7, and 8 keys. (The arrows on these keys are usually used for editing.)

Bear in mind that once a direction has been set the light box continues on its path until changed by the pressing of an arrow key. You must not go through any of the walls, no matter what. Just to liven up matters a little more, there is a time limit!



Important Note: In view of the fact that so much of the available memory is used up by this program, it is recommended that you type CLEAR and ENTER before you run it.

Program 15: FILL 'ER UP

```
1Ø PRINT AT 4,1Ø;"▛▀▀▀▀▀▀▀▀▀▜"
2Ø FOR I=5 TO 13
3Ø PRINT AT I,1Ø;"▌"
4Ø PRINT AT I,2Ø;"▐"
5Ø NEXT I
6Ø PRINT AT 14,1Ø;"▙▄▄▄▄▄▄▄▄▄▟"
7Ø DIM X(9,9)
8Ø LET R=1
```

```
 90 LET C=5
100 LET T=0
110 LET Y=2
120 GOTO 900
200 IF INKEY$<>"" THEN GOTO 240
210 GOSUB 1000
220 PAUSE 100
240 LET Y$=INKEY$
242 IF Y$="5" OR Y$="6" OR Y$="7" OR Y$=
    "8" THEN GOTO 248
245 GOTO 200
248 LET Y=VAL Y$
260 GOTO Y*100
500 LET C=C-1
510 IF C<1 THEN GOTO 1100
520 GOTO 900
600 LET R=R+1
610 IF R>9 THEN GOTO 1100
620 GOTO 900
700 LET R=R-1
710 IF R<1 THEN GOTO 1100
720 GOTO 900
800 LET C=C+1
810 IF C>9 THEN GOTO 1100
900 LET X(R,C)=1
910 PRINT AT R+4,C+10;"▒"
```

```
913 GOSUB 1ØØØ
92Ø IF INKEY$<>""THEN GOTO 2ØØ
93Ø FOR I=1 TO 9
94Ø FOR J=1 TO 9
95Ø IF X(I,J)=Ø THEN GOTO Y*1ØØ
96Ø NEXT J
97Ø NEXT I
98Ø PRINT AT 19,1Ø;"YOU DID IT"
99Ø STOP
1ØØØ LET T=T+1
1Ø1Ø IF T>13Ø THEN GOTO 11ØØ
1Ø2Ø RETURN
11ØØ IF T<13Ø THEN GOTO 115Ø
111Ø PRINT AT 19,12;"YOU_LOSE"
112Ø STOP
115Ø PRINT AT R+4,C+1Ø;"▒"
116Ø PRINT AT 19,1Ø;"BROKEN_WALL"
```

Points of Interest

Lines 10–60: The outer rim of the box is set up.

Lines 200–245: These accept one of the four arrow keys which control the direction taken by the moving line. None of the other keys has any effect on the program.

Line 248: Whatever key is used is then converted into a number in line 248, so that in line 260 it may be used in a calculated **GOTO**.

Lines 500–920: The filling up of the large box is taken care of by these lines.

Lines 930–990: These lines determine whether the area is completely filled in.

Lines 1000–1120: These lines affect the time limit.

Lines 1010–1100: The message that you have broken the wall is printed out, at which point the program terminates.

Possible Adaptations

If more than 2K memory is available, the frame may be enlarged to cover the entire screen. The **DIM** statement in line 70 would have to be changed accordingly. In addition, lines 610, 810, 930, and 940 would have to be amended, as well as the time limit in lines 1010 and 1100.

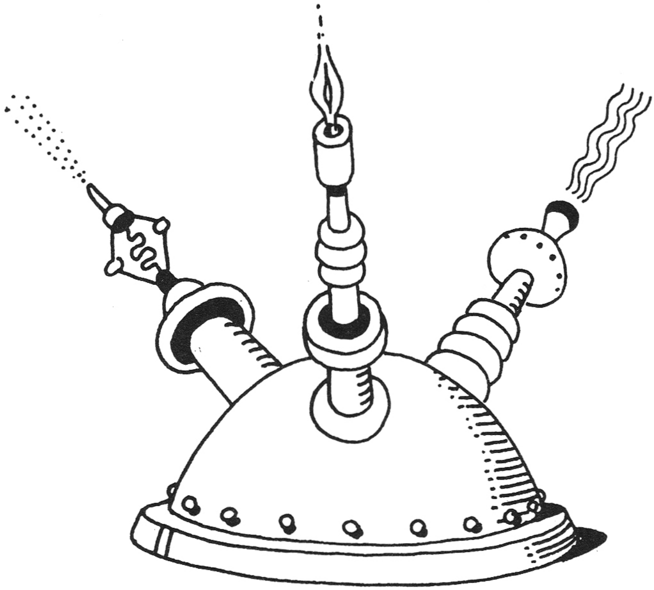
Chapter 16 SHOOT IT

- The Game
- Program 16: SHOOT IT
- Points of Interest
- Possible Adaptations

The Game

In order to get yourself in the right mood for this game, imagine that you are in your favorite shooting gallery—in outer space, of course. You are armed with three implosion beams and your mission is to blast away at a stream of stars that are orbiting around you in a menacing fashion. You may fire any one of the beams by typing the corresponding number 1, 2, or 3. However, there is the restriction that while one weapon is being fired the remaining two are temporarily disabled. The stars always travel in pairs, so that if you are fortunate enough to hit one you bag the other automatically.

In the game you are limited to a total of twenty shots. You not only have to gauge correctly the speed and the angle before you fire, you must also anticipate the future position of the stars. Admittedly this is not easy, but you are rewarded with no fewer than 250 points for each pair that is hit. Isn't it worth the effort?



Program 16: SHOOT IT

```
100 FOR I=7 TO 23 STEP 8
110 PRINT AT 18,I;"█"
120 PRINT AT 19,I-1;"▐";INT(I/7);"▌"
125 NEXT I
130 LET Z=0
135 LET P=0
140 LET X$="**__**__**__**__**__**__
    **__**__"
145 LET R=18
150 PRINT AT 1,8;"SHOOTING_GALLERY"
160 LET K=1
165 LET F=0
```

```
170 LET Y$=X$(1 TO 2)
180 LET X$(1 TO 30)=X$(3 TO 32)
190 LET X$(31 TO 32)=Y$
200 PRINT AT 2,0;X$
205 IF K=0 THEN GOTO 220
210 IF INKEY$<>"" THEN GOSUB 300
220 IF F=1 THEN GOSUB 330
230 IF Z=20 THEN GOTO 600
250 GOTO 170
300 IF INKEY$="1" THEN LET I=7
310 IF INKEY$="2" THEN LET I=15
320 IF INKEY$="3" THEN LET I=23
323 LET Z=Z+1
325 LET F=1
327 RETURN
330 LET R=R-1
340 PRINT AT R,I;"+"
350 PRINT AT R,I;"_"
360 LET K=0
370 IF R=2 THEN GOSUB 400
380 RETURN
400 IF X$(I)<>"_" THEN GOTO 450
410 LET F=0
420 LET K=1
430 LET R=18
440 RETURN
450 LET X$(I-1 TO I+1)="___"
```

```
46Ø LET P=P+25Ø
465 PRINT AT 21,14;P
47Ø LET K=1
48Ø LET R=18
49Ø LET F=Ø
5ØØ FOR J=1 TO 32
51Ø IF X$(J)<>"_" THEN RETURN
52Ø NEXT J
6ØØ PRINT AT 11,10;"END OF GAME"
```

Points of Interest

Lines 100–130: The **FOR/NEXT** loop in lines 100–130 sets up the three implosion beams at positions 7, 15, and 23.

Lines 150–250: These set the stars in motion.

Lines 300–320: These lines determine which weapon was fired: 1, 2, or 3.

Lines 330–380: The subroutine in lines 330–380 controls the motion of the stars and beams.

Lines 400–440: The subroutine in these lines checks if a hit has been made.

Lines 450–490: These lines tally the points scored and remove the damaged stars from the stream.

Lines 500–520: These lines determine whether there are any more targets left.

Possible Adaptations

1. The number of weapons can be increased from three. This would mean that line 100 would have to be changed as well as lines 300–320.

2. The stars could be made to travel faster. This would mean a change in lines 170–190. In order to increase the speed by a factor of 1.5, the following changes could be implemented:

```
170 LET Y$=X$ (1 TO 3)
180 LET X$ (1 TO 29)=(4 TO 32)
190 LET X$(30 TO 32)=Y$
```

3. The number of shots can be changed from twenty to any other number of your choosing.

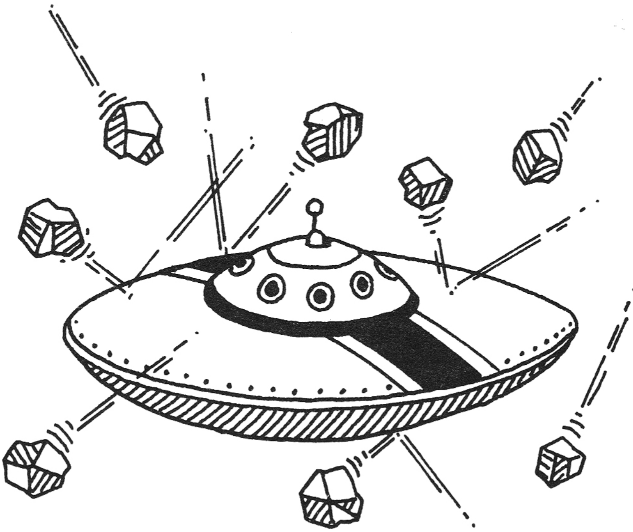
Chapter 17 RICOCHET

- The Game
- Program 17: RICOCHET
- Points of Interest
- Possible Adaptations

The Game

In this game you are being attacked by high-powered photon balls being aimed at you from inter-galactic attackers. You are in sole command of the five remaining earth stations. The attackers are unable to fire at you directly; they must fire at an orbiting satellite from which their shots are bounced off in your direction.

Whenever a photon ball is bounced off the satellite, it will head toward your five stations. It is then that you activate your defensive shields, with the proviso that at any given moment you can defend only one station. Keys 1,2,3,4, and 5 are used to activate your shields. If you succeed, your station is left intact and you amass additional energy, represented by gaining extra points. If you do not defend your station in time or you defend the wrong station, the station under attack will be demolished. In order for your complex shielding apparatus to work, you must keep at least



two stations intact. For this reason the game terminates when only one station remains.

Program 17: RICOCHET

```
 5  DIM A(5)
10  FOR I=5 TO 15
20  PRINT AT I,1;"■"
30  NEXT I
40  PRINT AT 1,10;"1_____2_____3_____4
    _____5"
45  LET H=0
50  PRINT AT 20,10;"1_____2_____3_____4
    _____5"
55  LET P=0
```

```
 6Ø LET X=INT (RND*5+1)
 7Ø GOSUB X*1ØØ+1ØØ
 75 LET J=(X-1)*5+1Ø
 8Ø FOR I=2 TO 1Ø
 9Ø PRINT AT I,J;"O"
1ØØ PRINT AT I,J;"_"
11Ø LET J=J+K
13Ø NEXT I
15Ø GOTO 65Ø
2ØØ LET K=-1
21Ø RETURN
3ØØ LET K=-1.5
31Ø RETURN
4ØØ LET K=-2
41Ø RETURN
5ØØ LET K=-2.5
51Ø RETURN
6ØØ LET K=-3
61Ø RETURN
65Ø LET X=INT (RND*5+1)
66Ø GOSUB X*1ØØ+1ØØ
67Ø LET J=2
68Ø FOR I=11 TO 19
69Ø PRINT AT I,J;"O"
7ØØ PRINT AT I,J;"_"
71Ø LET J=J-K
715 IF INKEY$<>"" THEN GOSUB 85Ø
```

```
72Ø NEXT I
73Ø PRINT AT 2Ø,(X-1)*5+1Ø;"_"
735 IF A(X)=1 THEN GOTO 81Ø
74Ø LET H=H+1
745 LET A(X)=1
75Ø IF H<4 THEN GOTO 81Ø
76Ø PRINT AT 12,12;"GAME OVER"
77Ø STOP
8ØØ LET P=P+5ØØ
81Ø PRINT AT 21,1Ø;"SCORE=_";P
82Ø GOTO 6Ø
85Ø LET Y$=INKEY$
86Ø IF Y$>="1" AND Y$<="5" THEN GOTO
    88Ø
87Ø GOTO 72Ø
88Ø IF VAL Y$=X AND A(X)=Ø THEN GOTO 8ØØ
89Ø GOTO 72Ø
```

Points of Interest

Lines 60–75: The angle at which the photon ball is aimed is controlled by lines 60–75.

Lines 200–610: The displacement for each of the 5 different angles of the photon balls is taken care of in lines 200–610. With a little help from line 710, this routine also handles the reflection angles.

Lines 730–745: These lines handle the elimination of any unprotected station and further ensure that the photon balls already fired will cause no further damage.

Lines 850–890: The routine in these lines checks to see whether the correct shields have been activated.

Possible Adaptations

If more than 2K memory is available, the speeds at which the photon balls are fired may be changed by modifying the following two lines:

```
80 FOR I=2 TO 10 STEP 2
```

and

```
680 FOR I=11 TO 19 STEP 2
```

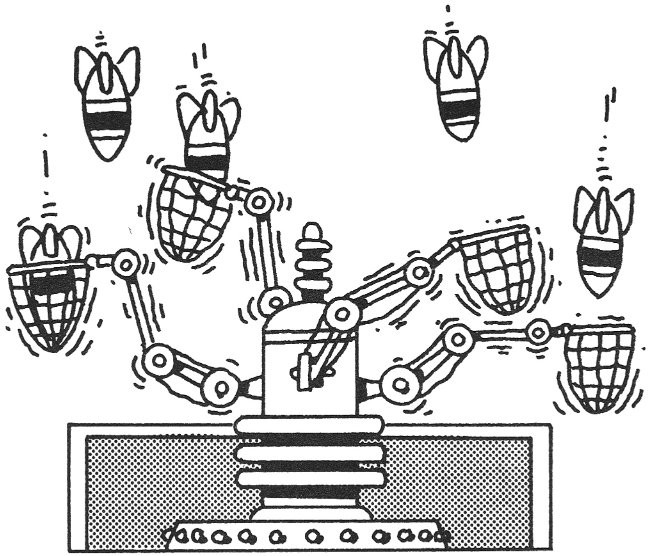
Chapter 18 INVASION

- The Game
- Program 18: INVASION
- Points of Interest
- Possible Adaptations

The Game

In this game, five different types of galactic aggressors have banded together and are planning to invade Earth. In order to pursue their aim, they dispatch small scout ships which gather information and relay it back to their mother ships. It is at that point that the scout ships glide to Earth in the form of guided missiles. Your mission as guardian of the Earth is to catch these falling ships and prevent them from crashing into major population centers. For this purpose you are provided with a super-catcher that can move to the left and to the right: use the 5 key for movement to the left and the 8 key for movement to the right. For rapid movement, use the H key, which will jettison you toward hyperspace.

If your aim is successful, you will observe the attacking ship trapped by your catcher, at which point part of your



catcher disappears temporarily. If you miss, you are debited 1 of the 5 "lives" you have.

It is only fair to warn you that the falling ships do not always fall in a straight line. They could fall diagonally or to the left or to the right. This is completely under the control of the computer.

Program 18: INVASION

```
 5Ø LET W=Ø
 6Ø LET R$="▒█▚▀▄"
 7Ø LET M=Ø
 8Ø LET J=15
 9Ø PRINT AT 2Ø,J;"▙▟"
1ØØ LET X=INT (RND*28)+2
1Ø5 LET T=INT (RND*5)+1
```

```
11Ø IF X<12 THEN GOTO 2ØØ
12Ø IF X>2Ø THEN GOTO 3ØØ
2ØØ LET Y=INT (RND*3)+1
21Ø IF Y=1 THEN LET K=Ø
22Ø IF Y>1 THEN LET K=.5
23Ø GOSUB 4ØØ
28Ø GOTO 1ØØ
3ØØ LET Y=INT (RND*3)+1
31Ø IF Y=1 THEN LET K=Ø
32Ø IF Y>1 THEN LET K=-.5
33Ø GOSUB 4ØØ
34Ø GOTO 1ØØ
4ØØ FOR I=1 TO 2Ø
41Ø PRINT AT I-1,X;"_"
42Ø LET X=X+K
43Ø PRINT AT I,X;R$(T)
435 IF INKEY$<>"" THEN GOSUB 5ØØ
44Ø NEXT I
45Ø PRINT AT I-1,X;"_"
46Ø IF J=X OR J+1=X THEN LET W=W+T*1ØØ
47Ø IF J<>X AND J+1<>X THEN LET M=M+1
48Ø IF M>=5 THEN GOTO 9ØØ
49Ø PRINT AT 21,7;"PTS=_";W;"__LIVES=_";
    5-M
495 RETURN
5ØØ LET Y$=INKEY$
51Ø IF Y$="5" THEN LET P=-1
```

```
52Ø IF Y$="8" THEN LET P=1
525 IF Y$="H" THEN GOTO 58Ø
53Ø IF J+P>3Ø OR J+P<1 THEN RETURN
54Ø PRINT AT 2Ø,J;"__"
55Ø LET J=J+P
56Ø PRINT AT 2Ø,J;"▙▟"
57Ø RETURN
58Ø PRINT AT 2Ø,J;"__"
59Ø LET J=INT (RND*3Ø)+1
6ØØ GOTO 56Ø
9ØØ PRINT AT 19,12;"GAME OVER"
91Ø PRINT AT 21,7;"PTS=_";W;"__LIVES=_";
    5-M
```

Points of Interest

Lines 100–120: These lines determine which ship will fall toward the earth.

Lines 200–340: These lines randomly set the path of the invading mother ship; the odds are good that it will fall in a diagonal direction.

Lines 460–495: The successes and failures are recorded in the routine between lines 460–495.

Lines 500–600: These handle the movement of the catcher as well as hyperspace. The hyperspace function randomly sets your catcher in a different location along the bottom of the screen.

Lines 900–910: The final results are displayed.

Possible Adaptations

You may wish to add to the 5 different shapes of the falling ships. This may be done by changing line 60. However, this change would have to be reflected in line 105 as well.

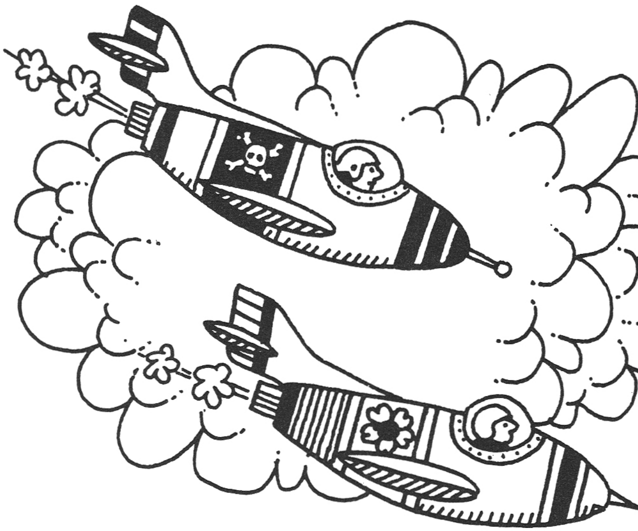
Chapter 19 DOGFIGHT

- The Game
- Program 19: DOGFIGHT
- Points of Interest
- Possible Adaptations

The Game

You are now a jet fighter pilot in the midst of the fiercest, wildest, most challenging duel of your entire career. The enemy jet is flying directly above you at twice your speed. You will recognize the enemy jet since it is shown in black on the screen, while yours is shaded gray. Every time the jets pass over each other, the enemy will try to riddle your jet full of holes. If you are quick you might be able to blast him before you are fired upon.

To fire a shot use the F key. If you hit the enemy jet it will be damaged and will retreat upwards on the screen, with you in hot pursuit. However, if the enemy hits your jet it will have the effect of causing your jet to fly at a lower level while he chases you. The game has two possible outcomes: either you succeed in driving the enemy away, in which case you become the master of the skies, or he manages to irreparably



damage your jet and force you to land, in which case, of course, you lose the game. Go at 'em!

Program 19: DOGFIGHT

```
100 LET E=8
110 LET P=11
120 LET F=3
125 LET X$="___"
130 LET Q=29
135 CLS
140 PRINT AT E,F;X$
150 PRINT AT E+1,F;X$
160 PRINT AT P,Q;X$
170 PRINT AT P+1,Q;X$
```

```
180 LET F=F+4
190 IF F>29 THEN LET F=3
200 LET Q=Q-2
210 IF Q<0 THEN LET Q=29
220 PRINT AT E,F;"▚▄▄"
230 PRINT AT E+1,F;"▞▀▀"
240 PRINT AT P,Q;"🮏🮏▟"
250 PRINT AT P+1,Q;"🮎🮎▜"
260 IF INKEY$<>"" THEN GOSUB 700
270 IF F<>Q THEN GOTO 140
275 PAUSE 10
280 IF INKEY$="F" THEN GOTO 400
290 PRINT AT P,Q+1;"+"
300 PRINT AT P+1,Q+1;"+"
310 PRINT AT 21,10;"YOU_WERE_HIT"
320 LET P=P+2
330 LET E=E+2
340 PAUSE 10
350 GOSUB 500
360 GOTO 135
400 PRINT AT E,F+1;"*"
410 PRINT AT E+1,F+1;"*"
420 PRINT AT 21,10;"YOU_HIT_HIM"
430 LET P=P-2
440 LET E=E-2
450 GOSUB 500
460 GOTO 135
```

```
5ØØ PAUSE 5Ø
51Ø IF E<1 THEN GOTO 55Ø
52Ø IF P>19 THEN GOTO 6ØØ
53Ø RETURN
55Ø PRINT AT 11,3;"YOU_DID_IT;_YOU_
    RULE_THE_SKY"
56Ø PRINT AT E+2,F;"_*_"
57Ø PRINT AT E+3,F;"_*_"
58Ø STOP
6ØØ PRINT AT 11,3;"YOUR_JET_WAS_
    FORCED_TO_LAND"
61Ø PRINT AT P-2,Q;"_+_"
62Ø PRINT AT P-1,Q;"_+_"
63Ø STOP
7ØØ PRINT AT P-1,Q+1;"*"
71Ø PRINT AT P-1,Q+1;"_"
72Ø RETURN
```

Points of Interest

Lines 140–250: The intricate jet movements are handled by lines 140–250. It is in this routine that the computer's jet is maintained at twice the speed of your jet.

Lines 260–280: These lines determine whether you are fast enough to hit him as he passes directly overhead.

Lines 310–340: Should your jet receive a direct hit, you will be forced to fly at a lower level. This eventuality is taken care of in lines 310–340.

Lines 400–450: The computer's jet is sent into retreat when you hit it. This is the role played by lines 400–450.

Lines 510–630: These lines determine if the game is over and, if so, who was victorious.

Possible Adaptations

You may want to change the speed of the jets (see lines 180, 190, 200, and 210). By changing line 275 you can also alter the interval of time allotted to fire the shots as the jets line up or meet. The **PAUSE** instruction can be amended for this change.

Chapter 20 SKETCH

- The Game
- Program 20: SKETCH
- Points of Interest
- Possible Adaptations

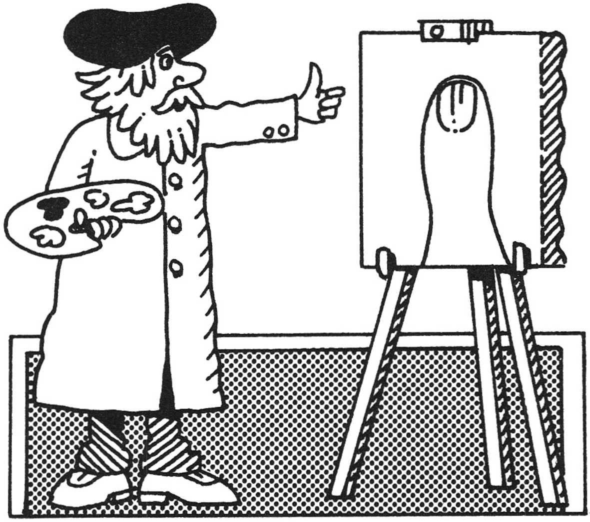
The Game

Don't be fooled by the brevity of this program. Despite the fact that it appears to be so short, it actually takes up most of the memory of the T/S 1000, mainly for setting up the display. The whole screen becomes your sketch pad. The "pen" or "brush" is the computer's plotter. You have the freedom to direct the pen to draw in any of the 4 directions indicated by the arrow keys (5,6,7 and 8). Here you have an opportunity to be as creative as you like.

Should you wish to terminate the movement of the pen at any point, merely press any key other than 5,6,7, and 8.

Program 20: SKETCH

```
1Ø LET I=Ø
2Ø LET J=Ø
```



```
 3Ø IF INKEY$<>"" THEN GOTO 5Ø
 5Ø IF INKEY$="5" THEN GOTO 1ØØ
 6Ø IF INKEY$="6" THEN GOTO 2ØØ
 7Ø IF INKEY$="7" THEN GOTO 3ØØ
 8Ø IF INKEY$="8" THEN GOTO 4ØØ
 9Ø GOTO 3Ø
1ØØ FOR X=I TO Ø STEP -1
11Ø PLOT X,J
12Ø LET I=X
14Ø IF INKEY$<>"" THEN GOTO 5Ø
15Ø NEXT X
16Ø GOTO 3Ø
2ØØ FOR Y=J TO Ø STEP -1
```

```
21Ø PLOT I,Y
22Ø LET J=Y
23Ø IF INKEY$<>"" THEN GOTO 5Ø
25Ø NEXT Y
26Ø GOTO 3Ø
3ØØ FOR Y=J TO 43
31Ø PLOT I,Y
32Ø LET J=Y
34Ø IF INKEY$<>"" THEN GOTO 5Ø
35Ø NEXT Y
36Ø GOTO 3Ø
4ØØ FOR X=I TO 63
41Ø PLOT X,J
42Ø LET I=X
44Ø IF INKEY$<>"" THEN GOTO 5Ø
45Ø NEXT X
46Ø GOTO 3Ø
```

Points of Interest

Lines 30–90: The keyboard readings are set up by these lines.

Lines 100–160: The movement of the pen to the left is taken care of by lines 100–160.

Lines 200–260: The downward pen movement is the responsibility of these lines.

Lines 300–360: Upward motion of the pen is controlled by these lines.

Lines 400–460: These move the pen to the right.

The careful reader of this program will notice that nowhere is there a single **PRINT** statement. In addition, the drawn line is thinner than any of the characters on the keyboard. How come? The answer lies in the fact that advantage is taken of this versatile computer's **PLOT** command.

Possible Adaptations

Should you choose, the character being drawn may be changed by replacing the **PLOT** command by the more conventional **PRINT AT**, which requires that you specify a particular character to be drawn. The user should bear in mind, however, that the **PRINT AT** instruction uses a screen size of 22 X 32 while **PLOT** uses a screen size of 44 X 64, and uses smaller pixels.

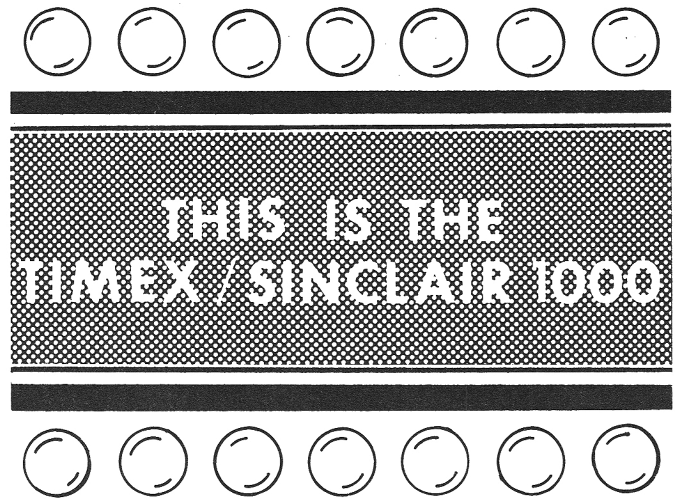
Chapter 21 MESG DISPLAY

- The Game
- Program 21: MESG DISPLAY
- Points of Interest
- Possible Adaptations

The Game

Most of us are familiar with the moving word displays used for advertising purposes or for news bulletins. This last program produces the same kind of display right on your screen. To start, respond to the question "What is your choice?" by typing in 0 and ENTER. When the computer asks you how many cycles you want, type in 3, again followed by ENTER. You should now see before your eyes the sample message: THIS IS THE T/S-1000... running across the screen in inverse video. It will do this 3 times.

Now that you've seen how it works, we can begin the fun part. In response to the question "What is your choice?" type in a 1. This allows you to input any message of your choosing, just so long as it is less than 32 characters. Whatever message you opt for will now be displayed across the screen for whatever number of cycles you type in. Whatever your message, you can put it up in lights!



Program 21: MESG DISPLAY

```
  5 DIM X$(32)
 1Ø PRINT "ENTER_A_Ø_FOR_MESSAGE"
 2Ø PRINT "ENTER_A_1_TO_INPUT_ONE"
 3Ø PRINT "WHAT_IS_YOUR_CHOICE?"
 4Ø INPUT X
 5Ø IF X=Ø THEN GOTO 11Ø
 6Ø PRINT "ENTER_YOUR_MESSAGE"
 7Ø INPUT X$
 8Ø LET L=LEN X$
 9Ø LET X$(L+1 TO 32)="_______________
    ______________________"
1ØØ GOTO 12Ø
```

```
110 LET X$="THIS IS THE T/S-1ØØØ ...
    ██████████"
12Ø PRINT "NO._OF_CYCLES?"
13Ø INPUT Y
135 CLS
14Ø FOR I=1 TO Y
15Ø FOR J=1 TO 32
16Ø LET Y$=X$(1)
17Ø LET X$(1 TO 31)=X$(2 TO 32)
18Ø LET X$(32)=Y$
19Ø PRINT AT 12,Ø;X$
2ØØ NEXT J
21Ø NEXT I
215 CLS
22Ø PRINT "ANY_MORE?_Y_OR_N"
23Ø INPUT Q$
24Ø IF Q$="Y" THEN GOTO 1Ø
25Ø PRINT AT 2Ø,12;"SO LONG"
```

Points of Interest

Lines 80 and 90: These lines contain a little trick that takes any character string you enter (less than 32 characters in length), and pads it with blanks so that it is actually 32 characters long.

Lines 150–200: The actual movement of the display is handled by the code in these lines. You might wish to incorporate this technique in some of your own programs in order to create a moving display.

Possible Adaptations

You may, if you desire, predefine several different messages and assign to them different number values. In this way the messages may be displayed "at the touch of a button." Should you have an important message that you want to be seen you could have it displayed on the screen in an otherwise darkened room. It would be most unlikely that such a message would be missed!

Chapter 22 SAVING YOUR PROGRAMS ON A TAPE RECORDER

- Saving a Program
- Loading a Program
- Important Reminders

Typing in a long program each time it is needed can be very time-consuming. Moreover, having to key in a program each time leaves open the door to error. However, a program can be saved on magnetic tape in much the same way that you ordinarily save a piece of music by recording it on your tape recorder.

Saving a Program

To save a program after it has been typed into your computer and checked out to be correct, all you have to do is to type the word **SAVE** and decide upon a name for your particular program. If you choose the name "COPY CAT," for example, you would type into your computer:

```
SAVE "COPY CAT"
```

At this point you should start the recorder in its "recording" mode. Then hit the ENTER key, and the program (or more accurately, a copy of your program, since

the program will still reside in the memory of the computer) will be recorded on the tape. You will know when the recording is complete by the termination code 0/0 which will appear at the bottom left-hand corner of the screen. At that point you may turn off the recorder.

Loading a Program

In order to load a program from magnetic tape into the computer, type in **NEW** followed by the ENTER key. Now you type either:

LOAD "COPY CAT"

which searches for and loads the program called "COPY CAT," or:

LOAD ""

which searches for and loads the first complete program found on the tape.

Important Reminders

1. Don't forget to make a note of the *precise* name of the program you are storing, including any spaces between words that may exist in the name. If you save a program called "COPY CAT" on tape and then later try to load a program called "COPYCAT" from the same tape, the computer will *not* be able to find the program.

2. We have found that for either loading or saving a program, a tape recorder in the "mono" (rather than "stereo") mode works best. If your tape recorder is stereo only, you should turn the "balance" control all the way left or all the way right. In addition, you should leave the "volume" control quite high, perhaps three-quarters of the way to full volume; the "tone" control should likewise be at its highest (treble) setting.

Appendixes

Appendix A KEYWORDS FOR THE T/S 1000

The following words, listed in alphabetical order, can be accessed on the T/S 1000 with a single keystroke. These keywords always appear in **BOLDFACE** in the program listings in this book.

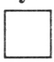
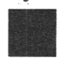
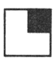
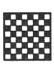
Keyword	**Mode**	**Key**	**Meaning**
ABS	F	G	absolute value
ACS	F	S	arcossine
AND	K or L	shifted-2	joining conditions
ASN	F	A	arcsine
AT	F	C	specify print position
ATN	F	D	arctangent
CHR$	F	U	number to character
CLEAR	K	X	clears variables and screen
CLS	K	V	clears screen
CODE	F	I	character to number
CONT	K	C	continue execution
COPY	K	Z	copies screen to printer
COS	F	W	cosine
DIM	K	D	dimensions arrays and strings
EXP	F	X	exponential
FAST	K or L	shifted-F	fast mode
FOR	K	F	sets up loop
GOSUB	K	H	subroutine branch
GOTO	K	G	unconditional branch
IF	K	U	condition testing
INKEY$	F	B	keyboard read
INPUT	K	I	accepts data
INT	F	R	returns integer portion

Keyword	Mode	Key	Meaning
LEN	F	K	length of string
LET	K	L	assigning values
LIST	K	K	lists program
LLIST	K or L	shifted-G	lists on printer
LN	F	Z	natural log
LOAD	K	J	loads program from tape
LPRINT	K or L	shifted-S	prints to printer
NEW	K	A	clears memory
NEXT	K	N	indicates end of loop
NOT	F	N	negation
OR	K or L	shifted-W	two or more conditions
PAUSE	K	M	pauses the machine
PEEK	F	O	shows memory at specified point
PI	F	M	value of π
PLOT	K	Q	places pixel on screen
POKE	K	O	places data in specified point of memory
PRINT	K	P	prints item
RAND	K	T	indicates starting position of **RND**
REM	K	E	remarks
RETURN	K	Y	return to **GOSUB** call
RND	F	T	random function
RUN	K	R	runs program
SAVE	K	S	saves to tape
SCROLL	K	B	scrolls one line
SGN	F	F	sign of a number
SIN	F	Q	sine
SLOW	K or L	shifted-D	slow mode
SQR	F	H	square root

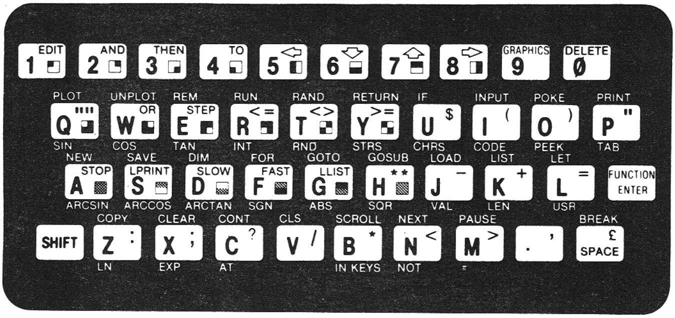
Keyword	Mode	Key	Meaning
STEP	K or L	shifted-E	increment of **FOR/NEXT**
STOP	K or L	shifted-A	stops execution
STR$	F	Y	numeric values to string
TAB	F	P	tabs over spaces
TAN	F	E	tangent
THEN	K or L	shifted-3	resultant of TRUE condition
TO	K or L	shifted-4	specifies boundaries
UNPLOT	K	W	unplots pixel
USR	F	L	machine code within BASIC
VAL	F	J	converts numeric characters to actual numeric values

Appendix B GRAPHICS SYMBOLS FOR THE T/S 1000

The following graphics symbols are available on the T/S 1000. Please note that in this listing each symbol is "boxed," meaning that all four of its borders are shown even though these borders will *not* appear on the screen. (For example, the very first symbol listed here will show up on the screen merely as a space, and not an empty box.) It should also be noted that letters in inverse video are also accessed via the **G** mode.

Symbol	Mode	Key	Symbol	Mode	Key
	**K** or **L**	SPACE		**G**	SPACE
	**G**	shifted-1		**G**	shifted-Q
	**G**	shifted-2		**G**	shifted-W
	**G**	shifted-7		**G**	shifted-6
	**G**	shifted-4		**G**	shifted-R
	**G**	shifted-5		**G**	shifted-8
	**G**	shifted-T		**G**	shifted-Y
	**G**	shifted-E		**G**	shifted-3
	**G**	shifted-A		**G**	shifted-H
	**G**	shifted-D		**G**	shifted-G
	**G**	shifted-S		**G**	shifted-F

Appendix C THE T/S 1000 KEYBOARD



Appendix D AN EXTRA COPY OF PROGRAM 1, COPY CAT

This is an extra copy of Program 1, "Copy Cat." You may tear it out of this book and mark it with your own reminders as you follow the "Keyboarding Notes" in Chapter 1.

Program 1: COPY CAT

```
1ØØ LET X$="ASDFGHJKL"
1Ø5 LET T=Ø
11Ø LET N=Ø
12Ø PRINT AT 3,12;"COPY CAT"
125 LET Y$=X$(INT (RND*9+1))
13Ø LET N=N+1
14Ø PRINT AT 11,15;Y$
15Ø PAUSE 75-N
16Ø PRINT AT 11,15;"_'
17Ø IF INKEY$=Y$ THEN GOTO 2ØØ
18Ø PRINT AT 19,5;"YOU_MISSED;_IT_WAS_
    (";Y$;")"
19Ø STOP
2ØØ LET T=T+1
21Ø PRINT AT 21,15;T
22Ø GOTO 125
```

Computers

CRUNCHERS

21 Simple Games for the Timex/Sinclair 1000 2K

COPYCAT ◆ RANDOM ◆ GATES ◆ SLOT-M ◆ HIGH ROLLER ◆ BACCARAT ◆ SPACE PROBE ◆ MATH QUIZ ◆ HANG 'EM ◆ SEQUENCE? ◆ SCRAMBLE ◆ CONCENTRATION ◆ ALIEN GUNNER ◆ BARRIERS ◆ FILL 'ER UP ◆ SHOOT IT ◆ RICOCHET ◆ INVASION ◆ DOGFIGHT ◆ SKETCH ◆ MESG

Yin Chiu ◆ Henry Mullish

Here are twenty-one compact and fully explained games programs written *expressly* for the Timex/Sinclair 1000, the 2K-RAM machine that has become one of the fastest-selling personal computers in the world. These games of chance and skill are written in BASIC and are all runnable with the 2K RAM (Random Access Memory) built into the T/S 1000. The games will not only intrigue your entire family, but provide an entertaining and informative introduction to programming. With only an ordinary television set and the T/S 1000, you can enter these programs and play away to your heart's content. And with the addition of a tape recorder, you can save the programs and load them back into the machine whenever you want to play one of the games or modify one of the listings.

The clearly written instructions that accompany each of the introductory programs will make it possible for the absolute beginner to enter and run the first game within one hour. And the notes and documentation that accompany the more advanced programs offer suggestions about how to alter the programs. Besides providing countless hours of entertainment, these twenty-one CRUNCHERS will serve as the ideal introduction to your T/S 1000.

Yin Chiu is a recent graduate of New York University. At present he is a programmer with a large New York bank.

Henry Mullish teaches computer science at New York University and is the author of more than fifteen books on computers.

BYTE BOOKS

McGraw-Hill

ISBN 0-07-010831-5